2

NAVAL POSTGRADUATE SCHOOL
Monterey, California

10996

THESIS

ADAPTIVE CONTROL FOR A SPACECRAFT ROBOTIC MANIPULATOR

by

George Janvier IV

December, 1993

Thesis Advisor:	Brij Agrawal
Co-Advisor	Roberto Cristi

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE 16 December 1993.	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE ADAPTIVE CONTROL FOR A SPACRAFT ROBOTIC MANIPULATOR.

5. FUNDING NUMBERS

6. AUTHOR(S) George Janvier IV

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE
A

13. ABSTRACT *(maximum 200 words)*

This research involves the development of an adaptive control law for a space based two-link robotic manipulator. Non-adaptive controllers are first obtained utilizing feedback linearization techniques. A direct adaptive controller is then developed through the linear parameterization of the system dynamics, and the implementation of a Kalman Filter based adaption law. The controllers are then simulated and compared for various levels of system parameter uncertainty. The adaptive controller is found to be superior to the non-adaptive controllers for high levels of system parameter uncertainty. The non-adaptive controller is found to compare favorably to the adaptive controller in some areas for low values of system parameter uncertainty. The non-adaptive controller is implemented experimentally, consistent with hardware constraints. Experimental results verify the need for adaptive control when system dynamics are present which have not been modelled.

14. SUBJECT TERMS adaptive control, robotic manipulators, spacecraft attitude dynamics and control

15. NUMBER OF PAGES 109

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

Adaptive Control for A Spacecraft Robotic Manipulator

by

George Janvier IV
Lieutenant, United States Navy
B.S., U.S. Naval Academy 1986

Submitted in partial fulfillment
of the requirements for the degrees of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

MASTER OF SCIENCE IN ENGINEERING SCIENCE (AERONAUTICS)
from the

NAVAL POSTGRADUATE SCHOOL
December 1993

Author: George Janvier IV
George Janvier IV

Approved by: Brij M. Agrawal
Brij Agrawal, Thesis Advisor

Roberto Cristi, Co-Advisor

Michael A. Morgan
Michael A. Morgan, Chairman Department of Electrical and Computer Engineering

Daniel J. Collins
Daniel J. Collins, Chairman Department of Aeronautics and Astronautics

ABSTRACT

This research involves the development of an adaptive control law for a space based two-lin robotic manipulator. Non-adaptive controllers are first obtained utilizing feedback linearization techniques. A direct adaptive controller is then developed through the linear parameterization of the system dynamics, and the implementation of a Kalman Filter based adaption law. The controllers are then simulated and compared for various levels of system parameter uncertainty. The adaptive controller is found to be superior to the non-adaptive controllers for high levels of system parameter uncertainty. The non-adaptive controller is found to compare favorably to the adaptive controller in some areas for low values of system parameter uncertainty. The non-adaptive controller is implemented experimentally, consistent with hardware constraints. Experimental results verify the need for adaptive control when system dynamics are present which have not been modelled.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

This work was made possible through the support and encouragement of many people. Professor Cristi provided the inspiration and background for the adaptive controller design. Professor Agrawal provided valuable guidance regarding the scope and focus of the research. Mr. Mike Louie and Mr. Rafford Bailey were very helpful in the experimental phase and provided the author with much needed reality checks. Professor Bang was a tremendous source of encouragement and was always available to answer even the most trivial questions. Most of all, thank you Terri, Christopher and Grace for your love and support throughout the whole process.

I. INTRODUCTION

A. BACKGROUND

Robotic systems have been utilized to perform a wide variety of functions for many years. Their speed, precision and reliability make them well suited for applications ranging from routine manufacturing processes to interplanetary space exploration. As mankind seeks to reach out to other worlds, the robot will play a crucial role.

Space based robotic systems are required to deal with some unique conditions which are not encountered by their terrestrial counterparts. The absence of a fixed base upon which to mount a robotic manipulator and the lack of significant external friction to dampen the system require special consideration by a control systems engineer. An effective control system must not only reposition the manipulator, but counteract the forces imparted on the main body by such a maneuver. This problem is further complicated when the space based manipulator is called upon to handle an object with unknown mass and inertia properties.

Robotic systems equations of motion can be developed through Lagrange's equations and are highly non-linear. Standard linear control techniques are not well suited to this kind of model. One approach to the control problem is to use feedback linearization. This technique involves the development of a linearizing controller to cancel system non-linearities. Linear control techniques are then applied to the linearized system.

When system parameter uncertainty is present, controller performance can be improved through the use of adaptive control in which the uncertain system parameters are estimated on-line. Research in adaptive control started in the early 1950's in connection with the design of autopilots foi high-performance aircraft. Practical applications in robotic control emerged in the 1980's. Initial research relied on restrictive assumptions or approximations including linearization of robot dynamics, decoupling assumption for joint motors and slow variation of the inertia matrix [Ref. 1] [Ref. 2][Ref. 3][Ref. 4]. Later research resulted in the linear parameterization of robot dynamics allowing the adaptive controller to fully account for the non-linear, time-varying and coupled nature of robot dynamics [Ref. 5] [Ref. 6] [Ref. 7].

Controllers for the NPS Spacecraft Robotic Simulator (SRS) were first developed by Sorenson [Ref. 8] and later modified by Yale [Ref. 9]. Both developments assumed perfect knowledge of robotic system parameters.

B. OVERVIEW AND OBJECTIVES

The focus of this research is to develop an adaptive controller for the NPS SRS and to experimentally verify non-adaptive controller characteristics. The equations of motion and control laws are developed in Chapter II. Chapter III contains a comparison of controller performances for varying levels of parameter uncertainty. Non-adaptive controller experimental implementation is discussed in Chapter IV. Chapter V includes a summary of the conclusions as well as topics for future research.

II. ANALYTICAL MODEL

The analytical model used in this research represents a spacecraft with a two-link manipulator attached. The manipulator is maneuvered by motors at its shoulder and wrist while a momentum wheel holds the spacecraft centerbody steady in a desired orientation. Motion of the system is confined to two dimensions and the spacecraft centerbody is allowed to rotate but not translate. These restrictions facilitate comparison between analytical and experimental results.

A. COORDINATE SYSTEMS

An overall system schematic is shown in Figure 1. This diagram presents the system coordinate frames used to develop the equations of motion. The coordinate frames utilized are the same as those utilized by Yale [Ref. 9:pp 7-9] and are illustrated in Figure 1. The centerbody and manipulator links are assumed to be rigid bodies. Therefore, member lengths(L_1,L_2), distances to centers of mass(L_{c0},L_{c1},L_{c2}), and the location where the manipulator attaches to the centerbody(L_0,θ_0) remain constant. An inertial axis system is located on the centerbody at the point of rotation. A body fixed coordinate frame uses the same origin as the inertial frame but rotates with the spacecraft centerbody. The x-axis of this frame points to the centerbody center of mass. The centerbody attitude, θ_0, is the angle between the inertial x-axis and the spacecraft centerbody x-axis. Each manipulator has its own set of body axes. A coordinate frame attached to the manipulator shoulder aligns its x-axis along the longitudinal axis of manipulator Link 1. The attitude of this link, θ_1, is zero when the link lies on a ray extending from the inertial origin through the shoulder. The attitude of Link 2 is defined by a coordinate frame attached to the elbow. The attitude of this link, θ_2, is zero when the link is parallel with Link 1. A set of generalized coordinates, $\underline{q}$, is chosen which describe the system include the centerbody attitude and joint angles for both manipulator links.

$$\underline{q} = [\theta_0 \theta_1 \theta_2]^T \tag{1}$$

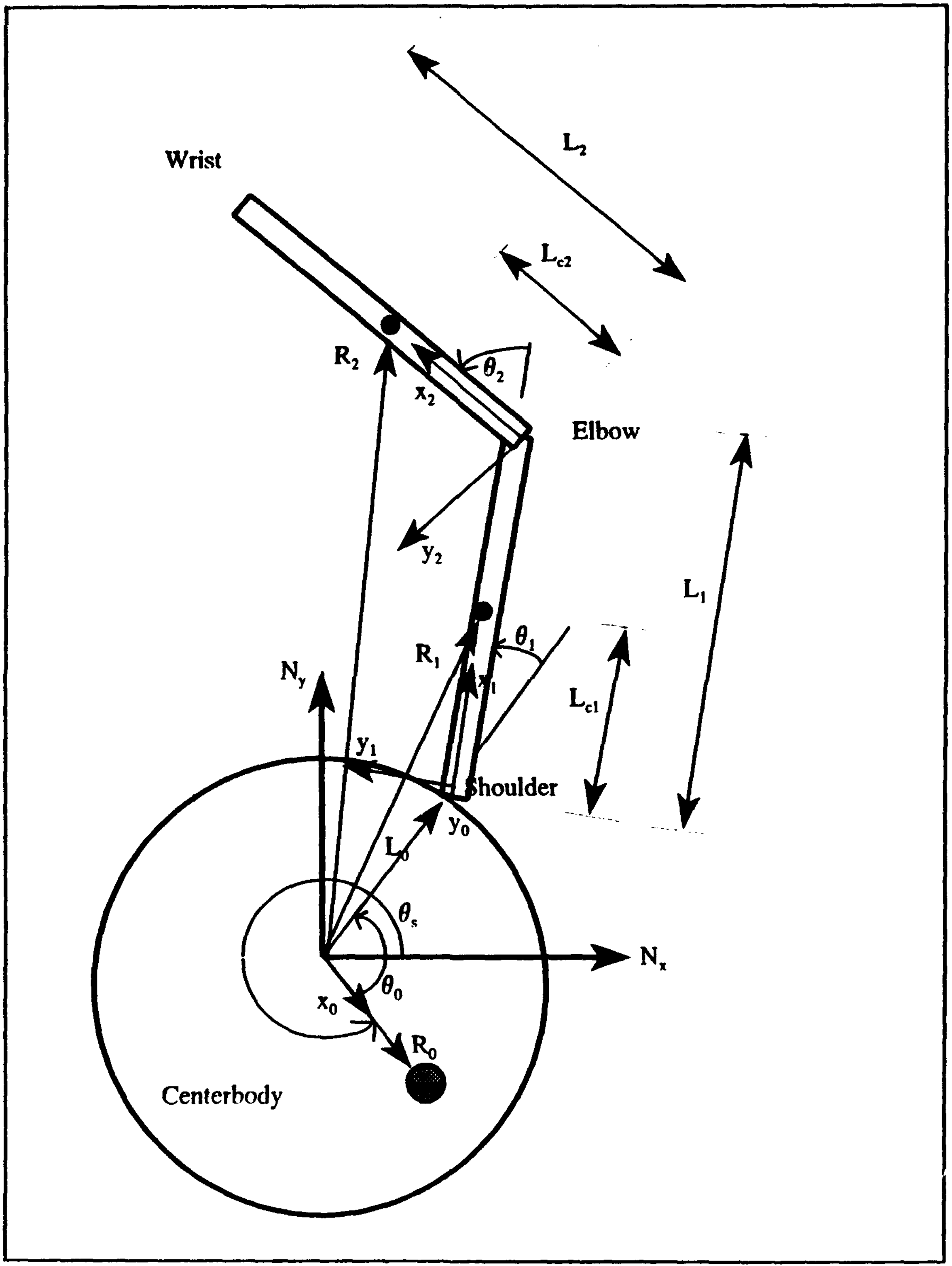

Figure 1: System Schematic

B. EQUATIONS OF MOTION

The equations of motion for this system are a special case of those developed by Yale [Yale93:pp. 9-24] and are derived using Lagrange's equations for a dynamic system.

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\underline{q}}}\right)-\frac{\partial L}{\partial \underline{q}}=\underline{Q} \qquad (2)$$

where:

- L=T-V;
- T is kinetic energy.
- V is potential energy.
- $\underline{q}$ is the generalized coordinates vector.
- $\underline{q}$ is the generalized velocities vector.
- $\underline{Q}$ is the vector of applied non-conservative forces.

Using Lagrange's equations, the equations of motion can be expressed in an alternate form.

$$M(\underline{q})\,\ddot{\underline{q}}+G(\underline{q},\dot{\underline{q}})+\frac{\partial V}{\partial \underline{q}}=\underline{Q} \qquad (3)$$

where:

- M is a 3x3 inertia matrix.
- $\underline{G}$ is a 3x1 vector which accounts for centripetal and Coriolis torques.
- V is the potential energy of the system.

Eq.(4) can be further simplified because the potential energy of the system is constant, and it becomes

$$M(\underline{q})\underline{\ddot{q}}+G(\underline{q},\underline{\dot{q}})=\underline{Q} \tag{4}$$

The following sections develop expressions for the inertia matrix, M,Coriolis vector, $\underline{G}$, and generalized force vector $\underline{Q}$.

1. Inertia Matrix, M

The inertia matrix is found by calculating the system kinetic energy and expressing it in the form

$$T=\frac{1}{2}\underline{\dot{q}}^{T}[M(\underline{q})]\underline{\dot{q}} \tag{5}$$

The total system kinetic energy, T, is the sum of the kinetic energy of all system components.

$$T=T_0+T_1+T_2 \tag{6}$$

Where T_0,T_1 and T_2 are the kinetic energies of the centerbody, Link 1 and Link 2 respectively. Kinetic energy of individual components can be found from

$$T_i=\frac{1}{2}I_i\omega_i{}^2+\frac{1}{2}m_i(\underline{\dot{r}}\cdot\underline{\dot{r}}) \tag{7}$$

where:

- I_i is the member moment of inertia about its center of mass.
- ω_i is the member angular velocity.
- m_i is the member mass.
- $\underline{\dot{r}}$ is the inertial velocity of the member center of mass.

Kinetic energies for individual members were determined by Yale [Yale93: pp 12-14] and are contained in Appendix A.

After substituting the expressions for kinetic energy obtained from Eqs.(6) and (7) into Eq.(4), the inertia matrix, M, is determined and is of the form

$$M=\begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} \tag{8}$$

Expressions for the individual elements in the inertia matrix are given by

$$M_{33}=I_2+m_2 l_{c2}^2 \tag{9}$$

$$M_{23}=M_{32}=M_{33}+m_2 l_1 l_{c2}\cos\theta_2 \tag{10}$$

$$M_{13}=M_{31}=M_{23}+m_2 l_0 l_{c2}\cos(\theta_1+\theta_2) \tag{11}$$

$$M_{22}=M_{23}+I_1+m_2 l_1 l_{c2}\cos\theta_2+m_1 l_{c1}^2+m_2 l_1^2 \tag{12}$$

$$M_{12}=M_{21}=M_{22}+l_0(m_1 l_{c1}+m_2 l_1)\cos\theta_1+m_2 l_0 l_{c2}\cos(\theta_1+\theta_2) \tag{13}$$

$$M_{11}=M_{22}+I_0+m_0 l_{c0}^2+(m_1+m_2)l_0^2+2l_0(m_1 l_{c1}+m_2 l_1)\cos\theta_1+2m_2 l_0 l_{c2}\cos(\theta_1+\theta_2) \tag{14}$$

2. Centripetal and Coriolis Vector, G

The Coriolis vector, G, contains all of the centripetal and Coriolis terms and may be found using

$$G(\underline{q},\dot{\underline{q}})=\begin{bmatrix}\dot{\underline{q}}^T C^{(1)}\dot{\underline{q}}\\ \dot{\underline{q}}^T C^{(2)}\dot{\underline{q}}\\ \dot{\underline{q}}^T C^{(3)}\dot{\underline{q}}\end{bmatrix} \tag{15}$$

where the elements $C^{(i)}$ are defined by the Christoffel symbol [Yale93: p 17]

$$C_{jk}^{(i)}=\frac{1}{2}\left(\frac{\partial M_{ij}}{\partial q_k}+\frac{\partial M_{ik}}{\partial q_j}+\frac{\partial M_{jk}}{\partial q_i}\right) \tag{16}$$

The $\underline{G}$ vector for the system is of the form

$$G=[G_1 G_2 G_3]^T \tag{17}$$

Expressions for individual elements of the G vector are given by

$$\begin{aligned}G_1=&-l_0(\dot\theta_1^2+2\dot\theta_0\dot\theta_1)(m_1 l_{c1}+m_2 l_1)\sin\theta_1-m_2 l_1 l_{c2}\dot\theta_2(2\dot\theta_0+2\dot\theta_1+\dot\theta_2)\sin\theta_2\\&-m_2 l_0 l_{c2}(2\dot\theta_0(\dot\theta_1+\dot\theta_2)+(\dot\theta_1+\dot\theta_2)^2)\sin(\theta_1+\theta_2)\end{aligned} \tag{18}$$

$$\begin{aligned}G_2=&l_0\dot\theta_0^2(m_1 l_{c1}+m_2 l_1)\sin\theta_1-m_2 l_1 l_{c2}\dot\theta_2(2\dot\theta_0+2\dot\theta_1+\dot\theta_2)\sin\theta_2\\&+m_2 l_0 l_{c2}\dot\theta_0^2\sin(\theta_1+\theta_2)\end{aligned} \tag{19}$$

$$G_3=m_2 l_1 l_{c2}(\dot\theta_1+\dot\theta_2)^2\sin\theta_2+m_2 l_0 l_{c2}\dot\theta_0^2\sin(\theta_1+\theta_2) \tag{20}$$

3. Generalized Forces, $\underline{Q}$

It is beneficial to express the vector of generalized forces, $\underline{Q}$, in terms of torque vector $\underline{U}$, representing torques applied by individual actuators. This is accomplished using the principle of virtual work. The total virtual work is the sum of the torques applied to each system component multiplied by each component's virtual angular displacement.

$$\delta W=\sum_{i=1}^{3}\delta W_i \tag{21}$$

The local torque vector is simply a 3 by 1 vector consisting of torques applied by the centerbody, shoulder, and elbow actuators respectively.

$$\underline{U}=[U_1 U_2 U_3]^T \tag{22}$$

Because the angles describing the system are expressed in local coordinates each actuator creates a virtual displacement only for its associated component.

$$\delta W_i=U_i\delta\theta_i \tag{24}$$

This yields the relation,

$$\underline{Q}=\underline{U} \tag{25}$$

4. Equations of Motion: Expressed in Local Coordinates

Substituting Eqs.(9)-(14),(18)-(20) and (25) into Eq.(4) produces the system equations of motion expressed as a function of local coordinates

$$M(\underline{\theta})\underline{\ddot{\theta}}+G(\underline{\theta},\underline{\dot{\theta}})=\underline{U} \tag{26}$$

Where,

$$\underline{\theta}=[\theta_0\theta_1\theta_2]^T \tag{27}$$

Eq.(26) is the equation of motion upon which system control laws are developed.

C. REFERENCE MANEUVER

Controller torques will be designed to cause the system to follow a desired reference trajectory. Given a three degree of freedom system, one needs only to specify the trajectory to be followed by three of the system's generalized coordinates. The remaining set of generalized coordinate trajectories can be found via geometric reasoning.

1. Selection of Reference Trajectory

The three coordinates chosen to be specified by the reference trajectory are the actuator tip x and y coordinates, R_x and R_y, and the centerbody angle θ_0. The generalized coordinates used in the system equations of motion, Eq.(26), are θ_0, θ_1 and θ_2. R_x and R_y can be expressed in terms of θ_0, ϑ_1 and θ_2.

The choices between reference trajectories which move the system from a given initial condition to a desired final condition are infinite. To help ensure that the spacecraft structure does not experience any unnecessary jerks or excitation of flexible structures, Sorenson utilized a fifth order polynomial with zero velocity and acceleration at initial and final conditions [Sore: pp 25-29].

$$f(\tau)=6\tau^5-15\tau^4+10\tau^3 \tag{28}$$

Where the normalized time, τ, is defined as

$$\tau=\frac{t-t_0}{t_f-t_0} \tag{29}$$

Given the desire to hold the centerbody attitude constant during a given manipulator maneuver, the centerbody angle, angular velocity and angular acceleration reference maneuvers are simply

$$\begin{aligned}\theta_0&=0\\ \dot{\theta}_0&=0\\ \ddot{\theta}_0&=0\end{aligned} \tag{30}$$

The manipulator tip position velocity and acceleration (R, $\dot{R}$, $\ddot{R}$) are found via

$$\begin{aligned}\underline{R}(\tau)&=\underline{R}(t_0)+f(\tau)\,[\underline{R}(t_f)-\underline{R}(t_0)]\\ \dot{\underline{R}}(\tau)&=\dot{f}(\tau)\left[\frac{\underline{R}(t_f)-\underline{R}(t_0)}{t_f-t_0}\right]\\ \ddot{\underline{R}}(\tau)&=\ddot{f}(\tau)\left[\frac{\underline{R}(t_f)-\underline{R}(t_0)}{(t_f-t_0)^2}\right]\end{aligned} \tag{31}$$

Where $\underline{R}(\tau)$ is the position vector originating at the centerbody point of rotation and ending at the tip of Link 2.

2. Coordinate Transformation

The system control laws to be developed will require angular position, velocity and acceleration of all elements of the system generalized coordinate vector $\underline{q}$. These can be found geometrically.

Generalized coordinates θ_0, R_x and R_y along with their respective velocities and accelerations have been derived in the previous section.

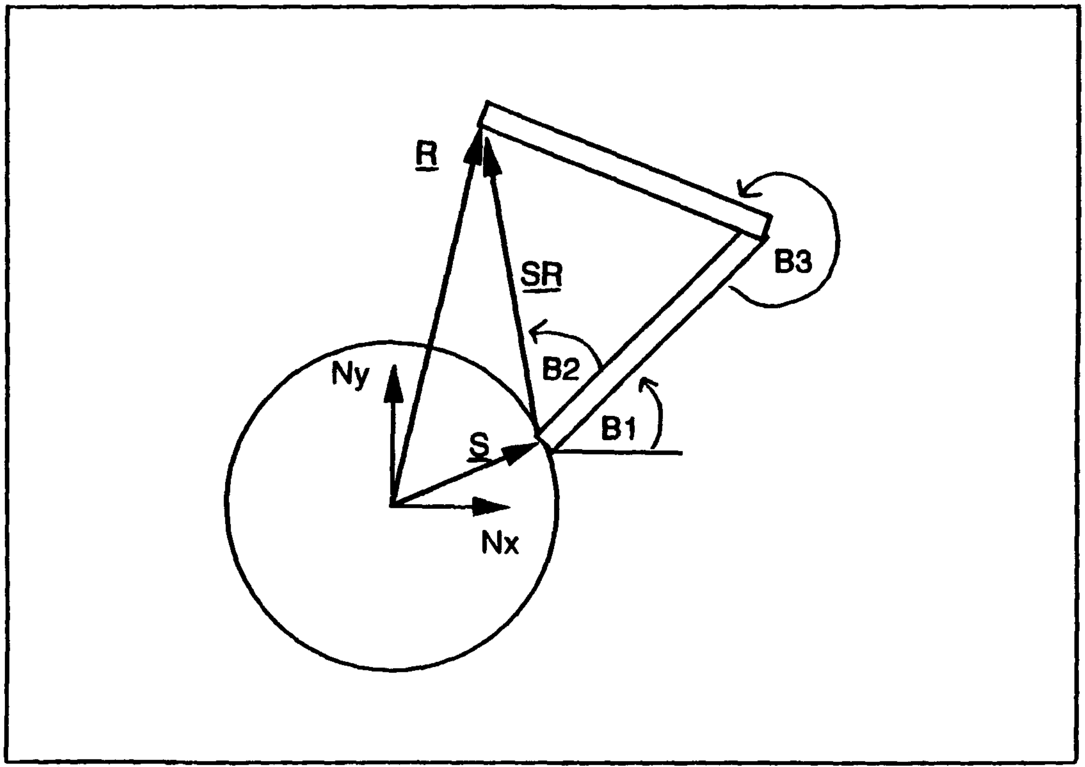

Figure 2: Manipulator Joint Angle Derivation Schematic

The position of the shoulder (S_x, S_y) and the magnitude of the vector between the shoulder and the tip of Link 2 (SR) can then be expressed as

$$\begin{aligned} S_x &= l_0\cos(\theta_0+\theta_s) \\ S_y &= l_0\sin(\theta_0+\theta_s) \\ SR &= \sqrt{(R_x-S_x)^2+(R_y-S_y)^2} \end{aligned} \tag{32}$$

Angles formed by the triangle formed by SR and the two manipulator links are found using the law of cosines.

$$\beta_1 = atan\left(\frac{R_y-S_y}{R_x-S_x}\right) \tag{33}$$

$$\beta_2 = acos\left(\frac{l_1^2+SR^2-l_2^2}{2l_1SR}\right) \tag{34}$$

$$\beta_3 = acos\left(\frac{l_1^2+l_2^2-SR^2}{2l_1l_2}\right) \tag{35}$$

The local angles are then found to be

$$\theta_1 = \beta_1-\beta_2-(\theta_0+\theta_s) \tag{36}$$

$$\theta_2 = 180^\circ-\beta_3 \tag{37}$$

Manipulator joint velocities and accelerations are found by expressing the manipulator end point coordinates in terms of system generalized coordinates.

$$R_x=l_0\cos(\theta_s+\theta_0)+l_1\cos(\theta_s+\theta_0+\theta_1)+l_2\cos(\theta_s+\theta_0+\theta_1+\theta_2) \qquad \textbf{(38)}$$

$$R_y=l_0\sin(\theta_s+\theta_0)+l_1\sin(\theta_s+\theta_0+\theta_1)+l_2\sin(\theta_s+\theta_0+\theta_1+\theta_2) \qquad \textbf{(39)}$$

Upon differentiation these equations can be expressed in the form

$$\begin{bmatrix}\dot{R}_x\\ \dot{R}_y\end{bmatrix}=H\begin{bmatrix}\theta_1\\ \theta_2\end{bmatrix} \qquad \textbf{(40)}$$

Where H is the Hamiltonian matrix expressed in the form

$$H=\begin{bmatrix}H_{11} & H_{12}\\ H_{21} & H_{22}\end{bmatrix} \qquad \textbf{(41)}$$

Individual elements of H are found to be

$$\begin{aligned}H_{11}&=-l_2\sin(\theta_s+\theta_0+\theta_1+\theta_2)-l_1\sin(\theta_s+\theta_0+\theta_1)\\ H_{12}&=-l_2\sin(\theta_s+\theta_0+\theta_1+\theta_2)\\ H_{21}&=l_2\cos(\theta_s+\theta_0+\theta_1+\theta_2)+l_1\cos(\theta_s+\theta_0+\theta_1)\\ H_{22}&=l_2\cos(\theta_s+\theta_0+\theta_1+\theta_2)\end{aligned} \qquad \textbf{(42)}$$

On the basis of Eq.(40)joint velocities are then found as

$$\begin{bmatrix}\theta_1\\ \theta_2\end{bmatrix}=H^{-1}\begin{bmatrix}\dot{R}_x\\ \dot{R}_y\end{bmatrix} \qquad \textbf{(43)}$$

together with the joint accelerations

$$\begin{bmatrix}\ddot{R}_x\\ \ddot{R}_y\end{bmatrix}=\dot{H}\begin{bmatrix}\theta_1\\ \theta_2\end{bmatrix}+H\begin{bmatrix}\theta_1\\ \theta_2\end{bmatrix} \qquad (44)$$

Rearranging Eq.(44) we obtain

$$\begin{bmatrix}\theta_1\\ \theta_2\end{bmatrix}=H^{-1}\left(\begin{bmatrix}\ddot{R}_x\\ \ddot{R}_y\end{bmatrix}-\dot{H}\begin{bmatrix}\theta_1\\ \theta_2\end{bmatrix}\right) \qquad (45)$$

3. Reference Torques

Given the reference values for the system generalized coordinates and their velocities and accelerations, a reference torque, $\underline{U}_r$, can be derived which would produce perfect tracking in the case of no external disturbances or modelling errors. The derived reference torque alone would represent an open loop type controller.

The system reference torque is derived by evaluating the system equations of motion at the reference values of the system generalized coordinates and their higher order derivatives as presented in Eq.(26).

D. NON-ADAPTIVE CONTROL LAW DESIGN

In this section two non-adaptive control laws are developed for the space based robotic manipulator. The first uses feedback linearization to cancel system non-linearities in conjunction with a PD type controller. The second is a modification to the first in which the non-linear portion of the controller utilizes generalized coordinate reference values instead of state feedback.

1. Linearizing Controller

a. Controller Design

The control law presented utilizes feedback linearization to cancel out non-linearities which occur in the system inertia matrix, M, and Coriolis vector, $\underline{G}$. A PD controller is then applied to the linearized system. The control law can be expressed in the following form

$$\underline{U}=\underline{U}_L-\delta\underline{U} \tag{46}$$

The linearizing term, $\underline{U}_L$, serves to cancel system non-linearities by predicting the current values of the system inertia matrix and Coriolis vector.

$$\underline{U}_L=M(\underline{\theta})\ddot{\underline{\theta}}_r+G(\underline{\theta},\dot{\underline{\theta}}) \tag{47}$$

The PD control term, $\delta\underline{U}$, corrects for tracking errors encountered by providing state feedback to the system.

$$\delta\underline{U}=M(\underline{\theta})[-K_v(\dot{\underline{\theta}}-\dot{\underline{\theta}}_r)-K_p(\underline{\theta}-\underline{\theta}_r)] \tag{48}$$

A controller block diagram is presented in Figure 3.

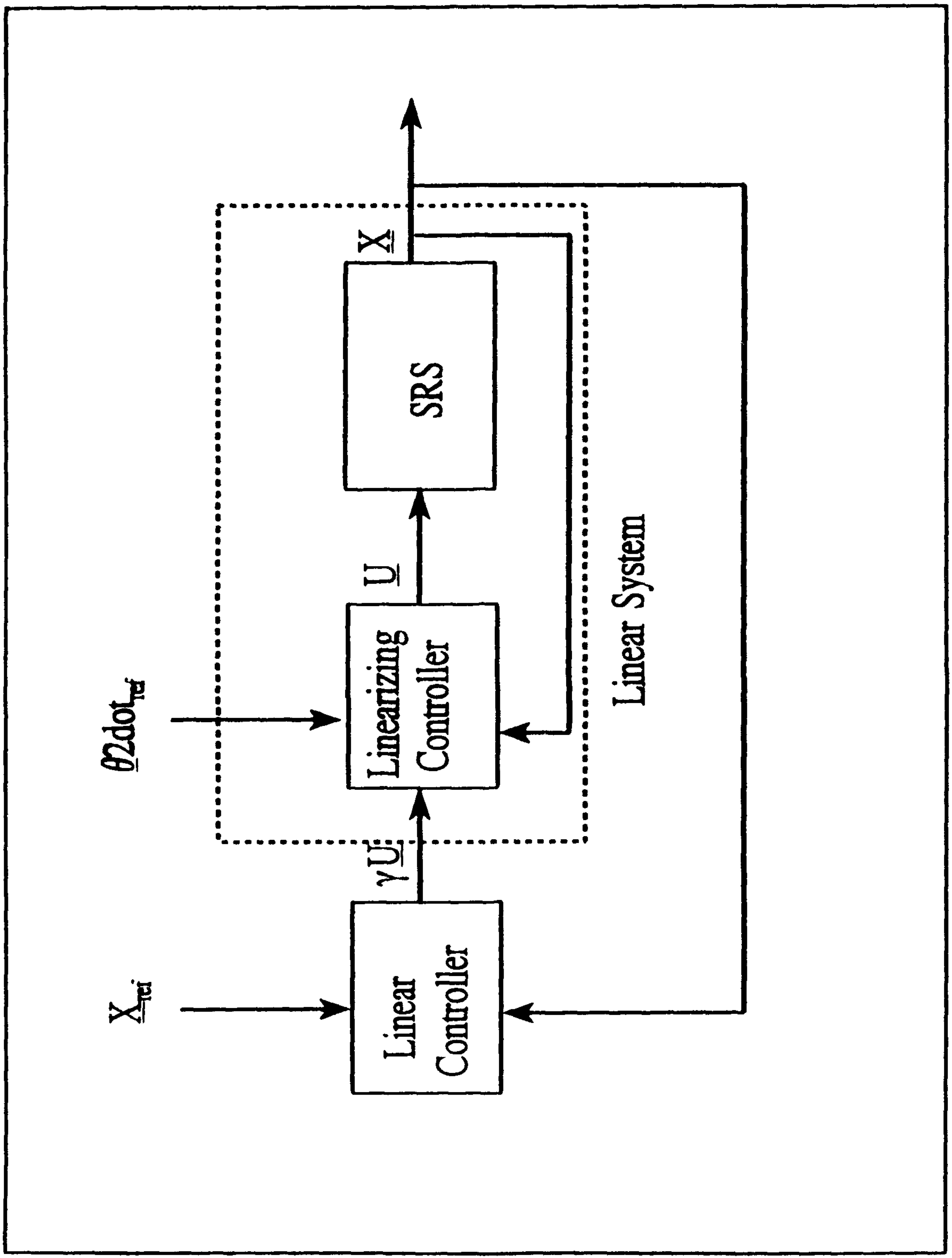

Figure 3: Linearizing Controller Block Diagram

b. Control Law Stability

Control law stability is determined by considering the behavior of the system trajectory error, $\underline{e}$. The trajectory error and its first and second derivatives are defined as

$$\underline{e}=\underline{\theta}-\underline{\theta}_r \qquad (49)$$

$$\underline{\dot{e}}=\underline{\theta}-\underline{\theta}_r \qquad (50)$$

$$\underline{\ddot{e}}=\underline{\theta}-\underline{\theta}_r \qquad (51)$$

Substituting Eqs,(49)-(51) into Eq.(46) yields

$$\underline{U}=G(\underline{\theta},\underline{\theta})+M(\underline{\theta})\,[\underline{\theta}_r-K_v\underline{\dot{e}}-K_p\underline{e}] \qquad (52)$$

which in turn can be combined with Eqs (50) and (25) to obtain

$$\underline{M}(\underline{\theta})\,(\underline{\ddot{e}}+K_v\underline{\dot{e}}+K_p\underline{e})=0$$

Because the system inertia matrix is positive definite and thus invertible, for any positive definite K_v and K_v

$$\lim_{t\to\infty}\underline{e}(t)=\underline{0}$$

2. Reference Controller

The reference controller presented is merely a slight modification to the linearizing controller presented previously. Rather than picking an inertia matrix and Coriolis vector to cancel out system non-linearities, M and $\underline{G}$ are calculated based on where the system should be as determined by a desired reference trajectory. The form of the controller is similar to that of the linearizing controller

$$\underline{U}=\underline{U}_r-\delta\underline{U} \tag{55}$$

The PD control term, $\delta\underline{U}$, is identical to that presented in Eq. (48). The linearizing term, $\underline{U}_r$, serves to cancel system nonlinearities by predicting the reference torques required to produce the desired reference trajectory when perfect tracking is assumed.

$$\underline{U}_r=M(\underline{\theta}_r)\ddot{\underline{\theta}}_r+G(\underline{\theta}_r,\dot{\underline{\theta}}_r) \tag{56}$$

A controller block diagram is presented in Figure 4.

E. ADAPTIVE CONTROL LAW DESIGN

Robotic manipulators are designed in order to grasp or manipulate an object. Often the mass and inertia properties of the object are not known beforehand. This in addition to modelling errors leads to uncertainty in system parameters and degraded control law performance. Adaptive control utilizes system input and output data to update the system parameters and thereby adjust to changes in system parameters.

1. Control Law Design

The adaptive controller developed is merely a modification to the non-adaptive reference controller presented in Eq.(55). The only difference between this controller and the non-adaptive version is a dependance on an

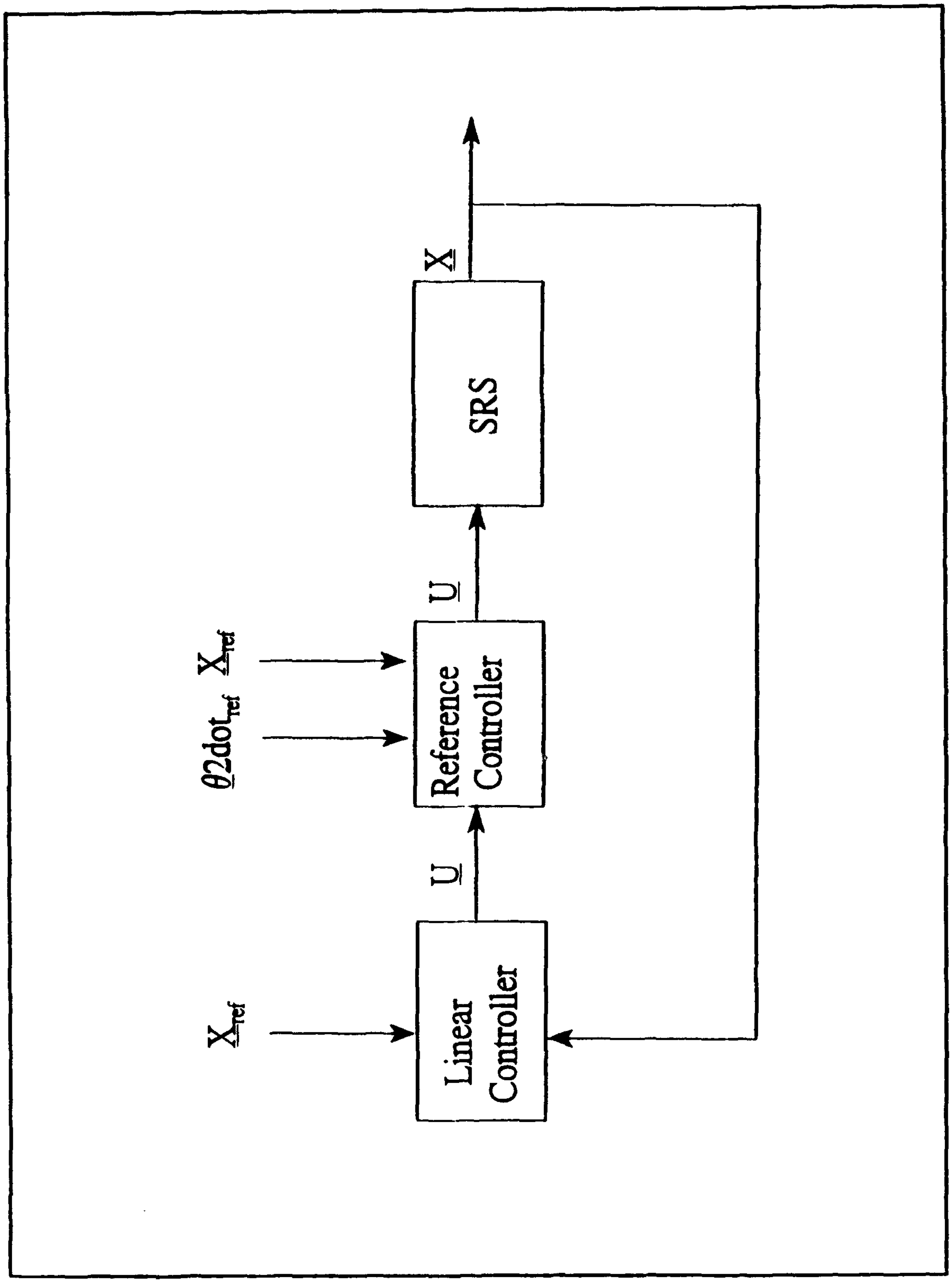

Figure 4: Reference Controller Block Diagram

estimate of the system parameter vector $\underline{A}$. In the non-adaptive case the system inertia matrix, M, and $\underline{G}$ vector are functions only of the generalized coordinates. In the adaptive case, M and $\underline{G}$ are functions of both generalized coordinates and the system parameter vector estimate $\underline{A}$.

$$\begin{gathered} M(\underline{\boldsymbol{\theta}},\hat{\underline{\boldsymbol{A}}}) \\ G(\underline{\boldsymbol{\theta}},\dot{\underline{\boldsymbol{\theta}}},\hat{\underline{\boldsymbol{A}}}) \end{gathered} \tag{57}$$

The meaning of $\underline{A}$ is developed in the following section. A controller block diagram is presented in Figure 4.

2. System Parameterization

The system parameter vector $\underline{A}$ is determined by expressing Eq.(25) in an alternate form

$$\boldsymbol{\Phi}^{T}(\underline{\boldsymbol{\theta}},\dot{\underline{\boldsymbol{\theta}}},\ddot{\underline{\boldsymbol{\theta}}})\hat{\underline{\boldsymbol{A}}}=\underline{u} \tag{58}$$

where $\boldsymbol{\Phi}$ is a function of the system generalized coordinates and $\underline{A}$ is a function of the system parameters. Equating Eqs.(26) and (58), $\underline{A}$ is found to be

$$\begin{aligned} &A(1)=I_2+m_2 l_{c2}^2 \\ &A(2)=m_2 l_1 l_{c2} \\ &A(3)=m_2 l_0 l_{c2} \\ &A(4)=I_1+m_1 l_{c1}^2+m_2 l_1^2 \\ &A(5)=l_0(m_1 l_{c1}+m_2 l_1) \\ &A(6)=I_0+m_0 l_{c0}^2+(m_1+m_2) l_0^2 \end{aligned} \tag{59}$$

The matrix of generalized coordinate $\boldsymbol{\Phi}$ is of the form

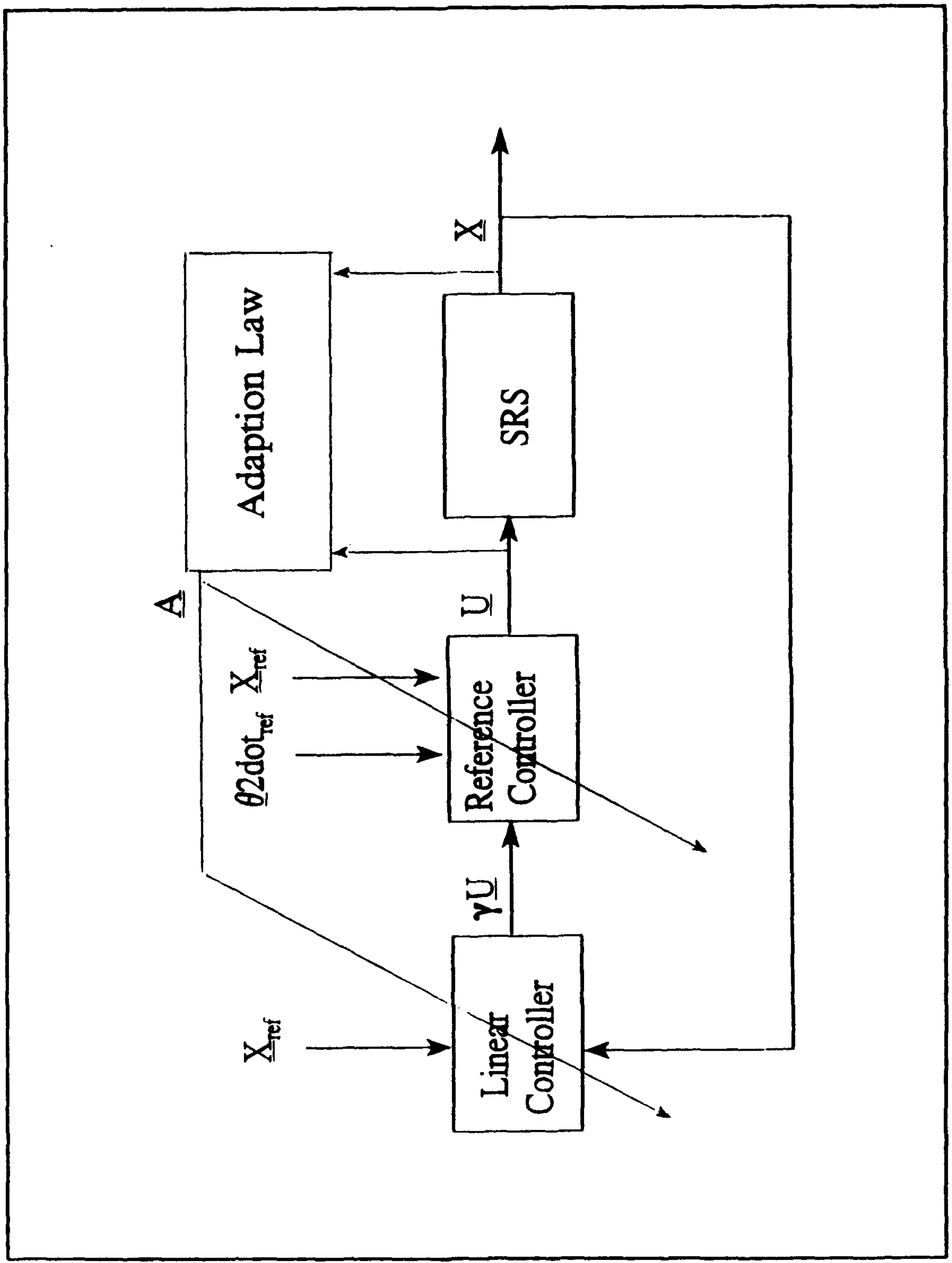

Figure 5: Adaptive Controller Block Diagram

$$\Phi=\begin{bmatrix}\Phi_{11} & \Phi_{12} & \Phi_{13}\\ \Phi_{21} & \Phi_{22} & \Phi_{23}\\ \Phi_{31} & \Phi_{32} & \Phi_{33}\\ \Phi_{41} & \Phi_{42} & 0\\ \Phi_{51} & \Phi_{52} & 0\\ \Phi_{61} & 0 & 0\end{bmatrix} \tag{60}$$

Expressions for the individual elements of Φ are given by

$$\Phi_{11}=\ddot{\theta}_0+\ddot{\theta}_1+\ddot{\theta}_2 \tag{61}$$

$$\Phi_{12}=\Phi_{11} \tag{62}$$

$$\Phi_{13}=\Phi_{11} \tag{63}$$

$$\Phi_{21}=(2\ddot{\theta}_0+2\ddot{\theta}_1+\ddot{\theta}_2)\cos\theta_2-\dot{\theta}_2(2\dot{\theta}_0+2\dot{\theta}_1+\dot{\theta}_2)\sin\theta_2 \tag{64}$$

$$\Phi_{22}=\Phi_{21} \tag{65}$$

$$\Phi_{23}=(\ddot{\theta}_0+\ddot{\theta}_1)\cos\theta_2+(\dot{\theta}_1+\dot{\theta}_2)^2\sin\theta_2 \tag{66}$$

$$\Phi_{31}=(2\ddot{\theta}_0+\ddot{\theta}_1+\ddot{\theta}_2)\cos(\theta_1+\theta_2)-(2\dot{\theta}_0(\dot{\theta}_1+\dot{\theta}_2)+(\dot{\theta}_1+\dot{\theta}_2)^2)\sin(\theta_1+\theta_2) \tag{67}$$

$$\Phi_{32}=\ddot{\theta}_0\cos(\theta_1+\theta_2)+\dot{\theta}_0^2\sin(\theta_1+\theta_2) \tag{68}$$

$$\Phi_{33}=\Phi_{32} \tag{69}$$

$$\Phi_{41}=\dot{\theta}_0+\dot{\theta}_1 \tag{70}$$

$$\Phi_{42}=\Phi_{41} \tag{71}$$

$$\Phi_{51}=(2\dot{\theta}_0+\dot{\theta}_1)\cos\theta_1-(\dot{\theta}_1^2+2\dot{\theta}_0\dot{\theta}_1)\sin\theta_1 \tag{72}$$

$$\Phi_{52}=\dot{\theta}_0\cos\theta_1+\dot{\theta}_0^2\sin\theta_1 \tag{73}$$

$$\Phi_{61}=\dot{\theta}_0 \tag{74}$$

3. Adaption Law

The system parameter vector, $\underline{A}$, is updated via a recursive Kalman Filter. The standard Kalman Filter state space equations can be expressed in the form

$$\begin{gathered}\underline{X}(t+1)=\sigma\underline{X}(t)+\Delta\underline{W}(t)\\ \underline{Y}(t)=C\underline{X}(t)+\underline{V}(t)\end{gathered} \tag{75}$$

Assuming a constant system parameter vector $\underline{A}$ for a given maneuver, the system can be expressed in state space form as

$$\begin{gathered}\underline{A}(t+1)=\underline{A}(t)+\underline{W}(t)\\ \underline{U}(t)=\Phi^{T}(t)\underline{A}(t)+\underline{V}(t)\end{gathered} \tag{76}$$

Equating Eqs.(75) and (76) yields the following relations.

$$\begin{aligned}&\underline{X}=\underline{A}\\&\sigma=\Delta=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}\\&\underline{y}=\underline{U}\\&\Phi^{T}=C\end{aligned}\tag{77}$$

Applying standard Kalman Filter equations to Eq.(76) yields the following set of recursive equations

$$\begin{aligned}&\underline{A}(t+1)=\underline{A}(t)+K(t)\,[\underline{u}(t)-\Phi^{T}(t)\underline{a}(t)]\\&K(t)=\overline{P}(t)\Phi(t)\,[\lambda+\Phi^{T}(t)\overline{P}(t)\Phi(t)]^{-1}\\&\overline{P}(t+1)=\overline{P}(t)-K(t)\Phi^{T}(t)\overline{P}(t)+\overline{Q}\end{aligned}\tag{78}$$

where

- K(t) is the Kalman Filter gain.
- λ is the noise covariance matrix, $E[\underline{e}(t)\ \underline{e}^{T}(t)]$.
- $\overline{P}$ is the parameter error covariance matrix, $E[(\underline{A}(t)-\underline{A}_{actual})(\underline{A}(t)-\underline{A}_{actual})^{T}]$.
- $\overline{Q}$ is the plant noise covariance matrix, $E[\underline{v}(t)\underline{v}^{T}(t)]$.

Because the noise covariance matrices, λ and Q, are not known, the parameter error covariance matrix, P, and the plant noise covariance matrix, Q, are redefined

$$\begin{aligned}&P(t)=\frac{\overline{P}(t)}{\lambda}\\&Q=\frac{\overline{Q}}{\lambda}\end{aligned}\tag{79}$$

Combining Eqs.(23) and (24) yields the recursive equations

$$
\begin{aligned}
&\underline{A}(t+1)=\underline{A}(t)+K(t)\left[\underline{u}(t)-\Phi^{T}(t)\underline{a}(t)\right]\\
&K(t)=P(t)\Phi(t)\left[I+\Phi^{T}(t)P(t)\Phi(t)\right]^{-1}\\
&P(t+1)=P(t)-K(t)\Phi^{T}(t)P(t)+\underline{Q}
\end{aligned}
\qquad (80)
$$

Eq.(80) provides the recursive equations necessary to update the system parameter vector, $\underline{A}$.

In the next chapter the three controllers are implemented for various levels of parameter uncertainty and their performances compared.

III. SIMULATION RESULTS

The computer simulations presented in this chapter were obtained using the MATLAB subroutines listed in Appendix B. Simulations are presented for various levels of system parameter uncertainty ranging from 0% to 500%. System parameter values presented in Table 1 correspond to actual values of the Spacecraft Robotics Simulator and are utilized to simulate system dynamics. Parameter values used by the controller contain a random error up to a specified percentage of the actual parameter value.

Equations of motion and computer code are verified by examining the change in angular momentum of the system for each simulation. For a given maneuver the rate of change in angular momentum will equal the sum of external torques on the system. The only external torque experienced by the Spacecraft Robotic Simulator is produced by the centerbody momentum wheel. Thus, for each simulation the relation

$$\dot{H}-U_{wheel}=0 \tag{77}$$

should be satisfied. Where H is the rate of change in angular momentum and U_{wheel} is the torque produced by the centerbody momentum wheel. The right hand side of Eq(77) was found to be $< 10^{-15}$ Nm/s^2 for all simulations.

A. SIMULATION TEST CASES

Simulation results are presented for five cases. The first case trains the adaptive control to recognize centerbody characteristics. Cases 2-6 examine the effects of parameter uncertainty on a desired manipulator maneuver. During the maneuver, the manipulator tip is repositioned from an initial to a final point along a straight line between the two points as shown in Figure 6. The angular position of the centerbody is held constant.

1. Case 1: Adaptive Controller Training Maneuver

Adaptive parameter $\underline{A}(6)$ depends only on centerbody characteristics. In order to update this parameter, a centerbody maneuver is required. Cases 2-6 attempt to hold the centerbody fixed and produces a small centerbody angular position, velocity and acceleration. A separate case, in which the centerbody is maneuvered, is required to adaptively update centerbody characteristics. During this maneuver, the manipulators are maneuvered in accordance with the reference maneuver pictured in Figure 6 while the centerbody is maneuvered as shown in Figure 7. Once the centerbody parameter,$\underline{A}(6)$, is updated, it is assumed fixed and no error is induced into this parameter in Cases 2-6.

TABLE 1: SYSTEM PARAMETER VALUES

Parameter Name	Parameter Variable	Parameter Value
Centerbody Radius	L_0	0.427 m
Arm 1 Length	L_1	0.530 m
Arm 2 Length	L_2	0.533 m
Centerbody CM	L_{c0}	0.104 m
Arm 1 CM	L_{c1}	0.403 m
Arm 2 CM	L_{c2}	0.314 m
Centerbody Mass	m_0	65.96 kg
Arm 1 Mass	m_1	2.34 kg
Arm 2 Mass	m_2	2.86 kg
Centerbody Inertia	I_0	5.74 kg-m^2
Arm 1 Inertia	I_1	0.081 kg-m^2
Arm 2 Inertia	I_2	0.182 kg-m^2

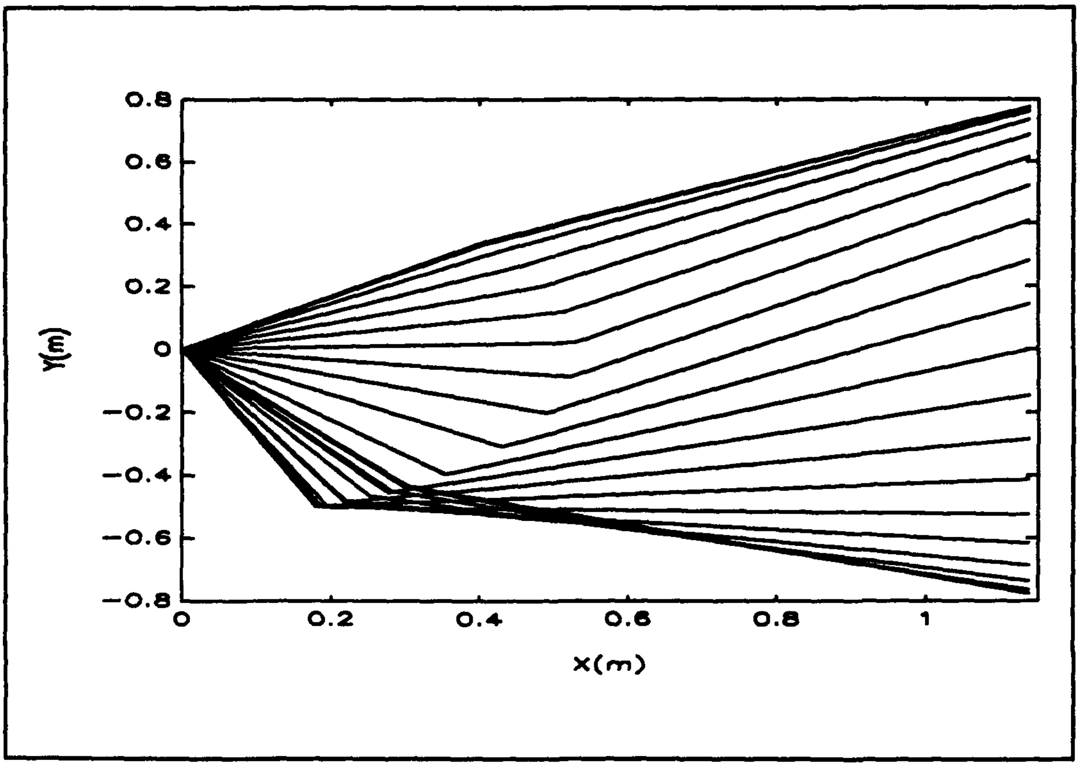

Figure 6: Reference Maneuver Time Lapse Stick Figure

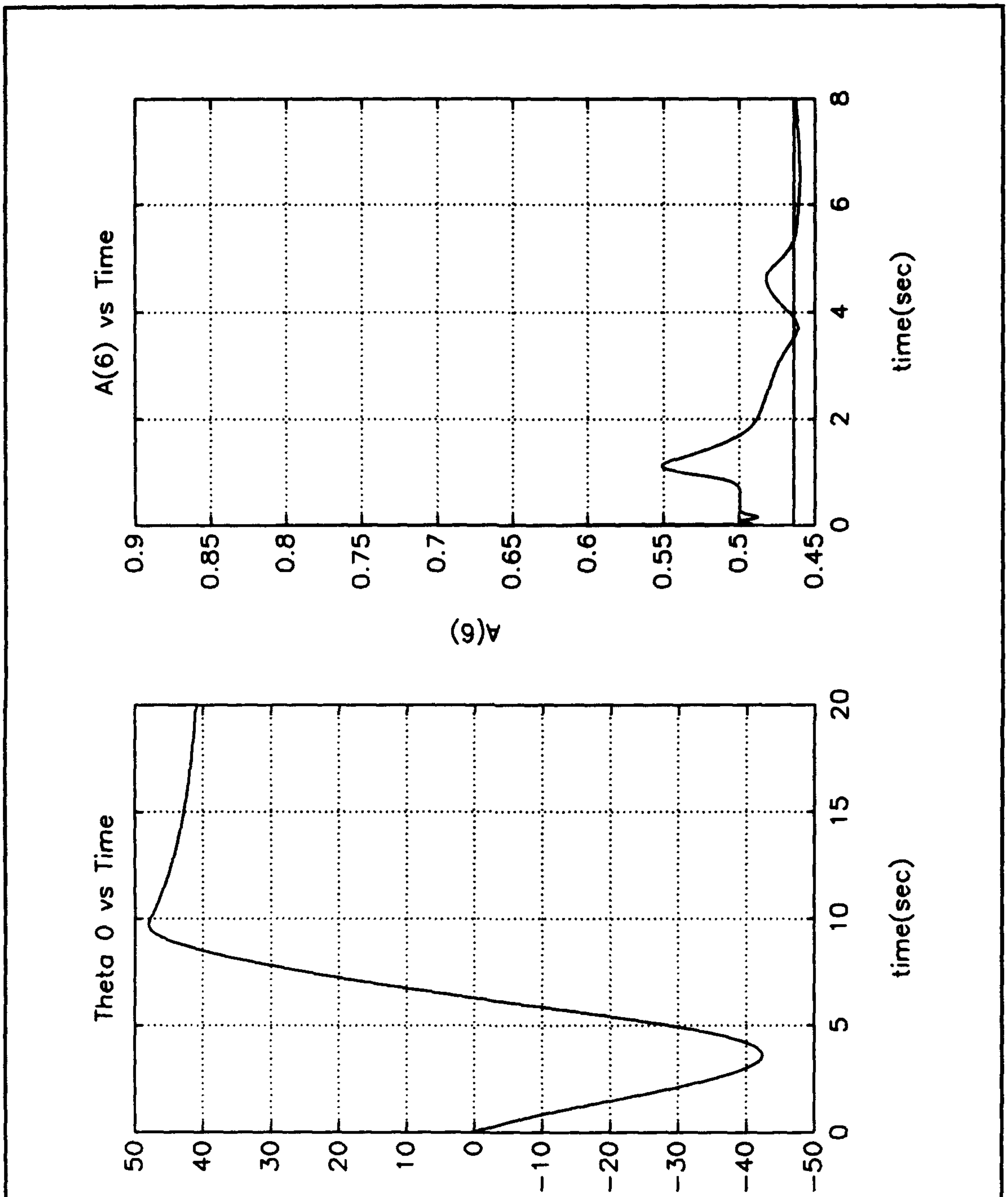

Figure 7: Centerbody Parameter Training Maneuver

2. Cases 2-6: 0-500% Parameter Uncertainty

Controller performance is presented for various levels of parameter uncertainty ranging from 0-500%. Controller errors and adaptive parameter updates are presented in Figures 8-23. Centerbody control torque characteristics are presented in Table 2.

B. COMPARISON OF CONTROLLERS

1. Adaptive Controller vs Non-adaptive Controller

The adaptive controller is clearly superior to the non-adaptive controllers for large values of parameter uncertainty (>50% parameter uncertainty). For small values of parameter uncertainty, the linearizing controller is superior to the adaptive controller in all but centerbody control.

2. Linearizing Controller vs Reference Controller

The linearizing controller is superior to the reference controller for low to moderate values of parameter uncertainty (<150% uncertainty). The reference controller exhibits superior performance over the linearizing controller for the case of 150% parameter uncertainty.

TABLE 2: CENTERBODY CONTROL TORQUE CHARACTERISTICS

Controller Type/ parameter uncertainty	U_{wheel} Max	U_{wheel} Min	$\int abs(U_{wheel})$
Linearizing (0%)	0.3735	-0.3987	0.2468
Reference (0%)	0.3735	-0.3987	0.2468
Linearizing (50%)	0.3723	-0.3987	0.2464
Reference (50%)	0.3738	-0.3980	0.2472
Adaptive (50%)	0.3735	-0.3987	0.2468
Linearizing (100%)	0.3741	-0.3963	0.2489
Reference (100%)	0.3742	-0.4000	0.2490
Adaptive (100%)	0.3735	-0.3987	0.2468
Linearizing (150%)	0.3736	-0.3958	0.2496
Reference (150%)	0.3744	-0.3930	0.2491
Adaptive (150%)	0.3734	-0.3987	0.2470
Adaptive (500%)	1.6784	-2.1698	0.2659

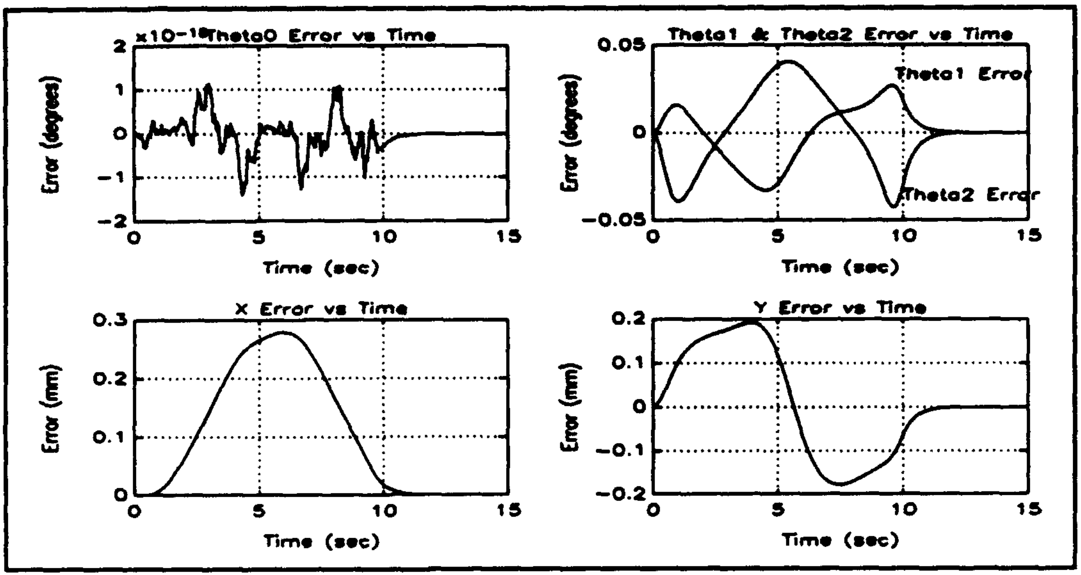

**Figure 8: Linearizing Controller Error
(0% parameter uncertainty)**

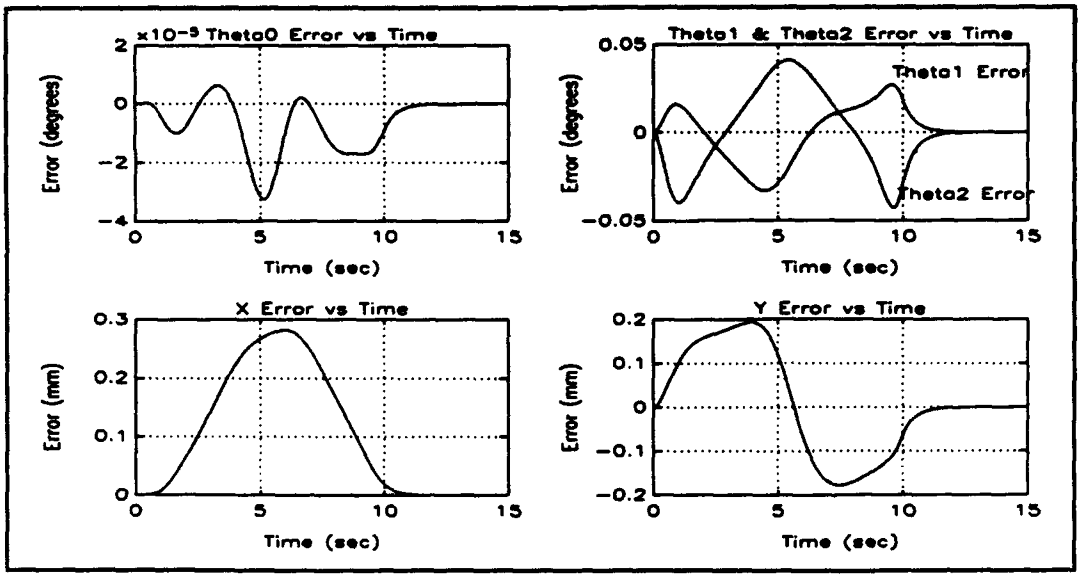

**Figure 9: Reference Controller Error
(0% parameter uncertainty)**

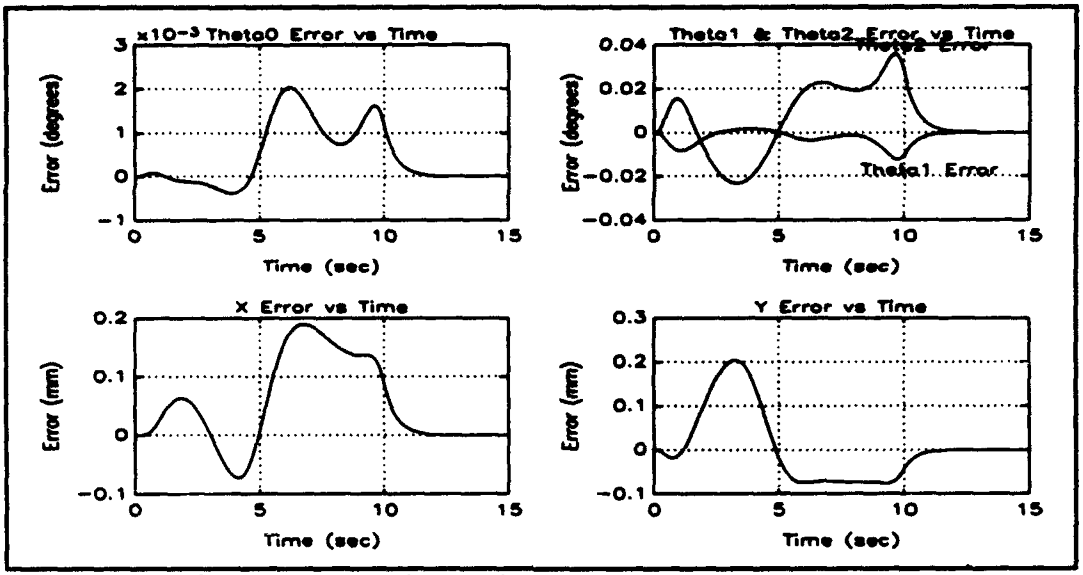

Figure 10: Linearizing Controller Error (50% parameter uncertainty)

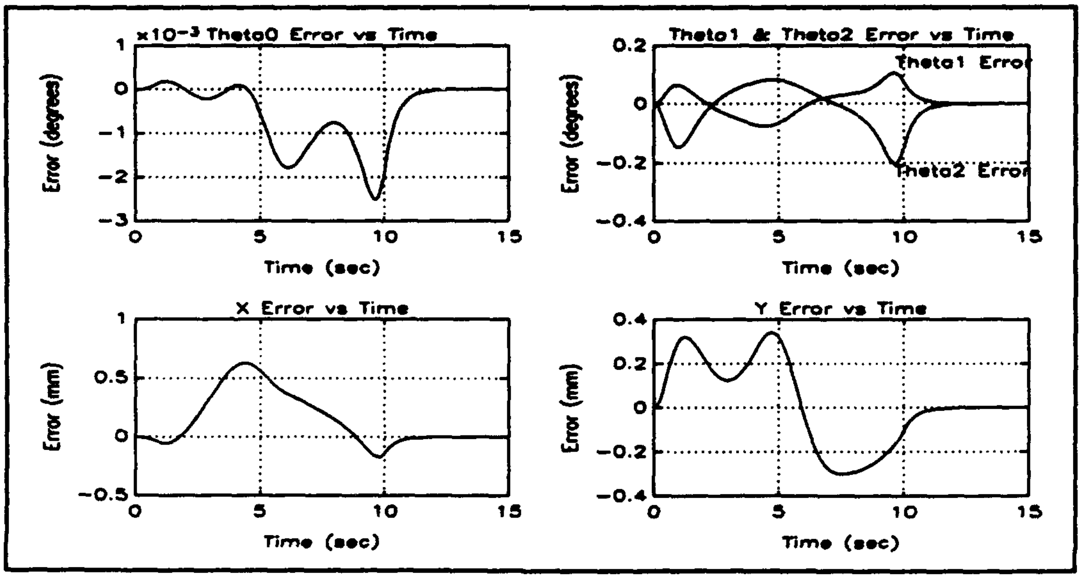

Figure 11: Reference Controller Error (50% parameter uncertainty)

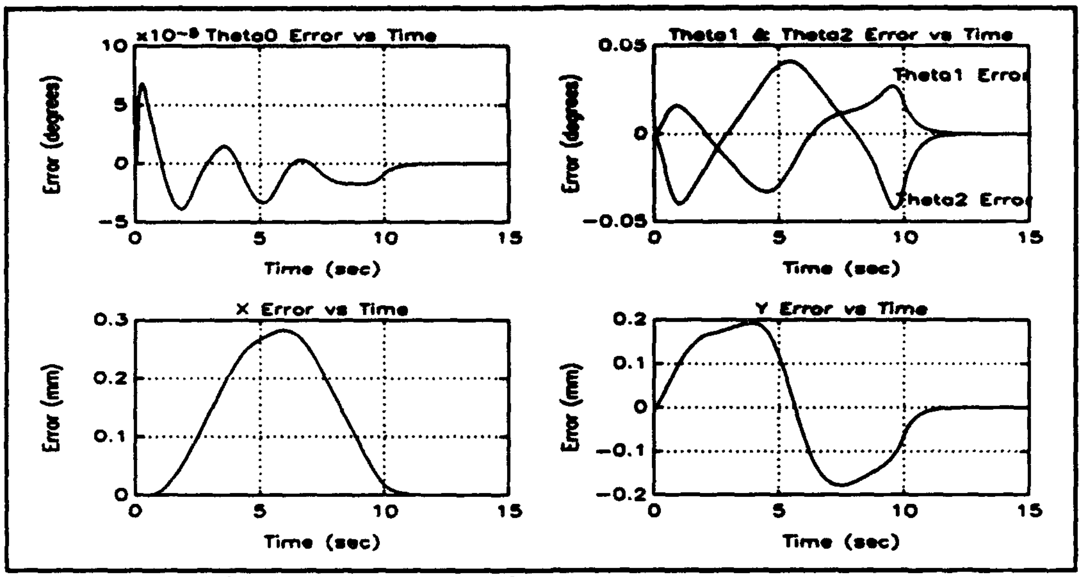

Figure 12: Adaptive Controller Error
(50% parameter uncertainty)

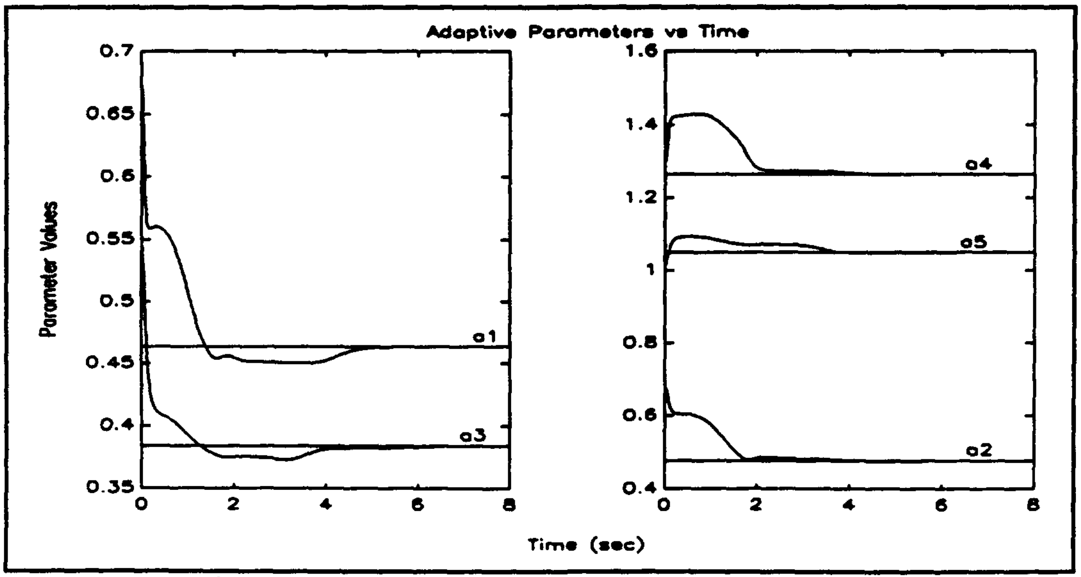

Figure 13: Adaptive Parameter Updates
(50% parameter uncertainty)

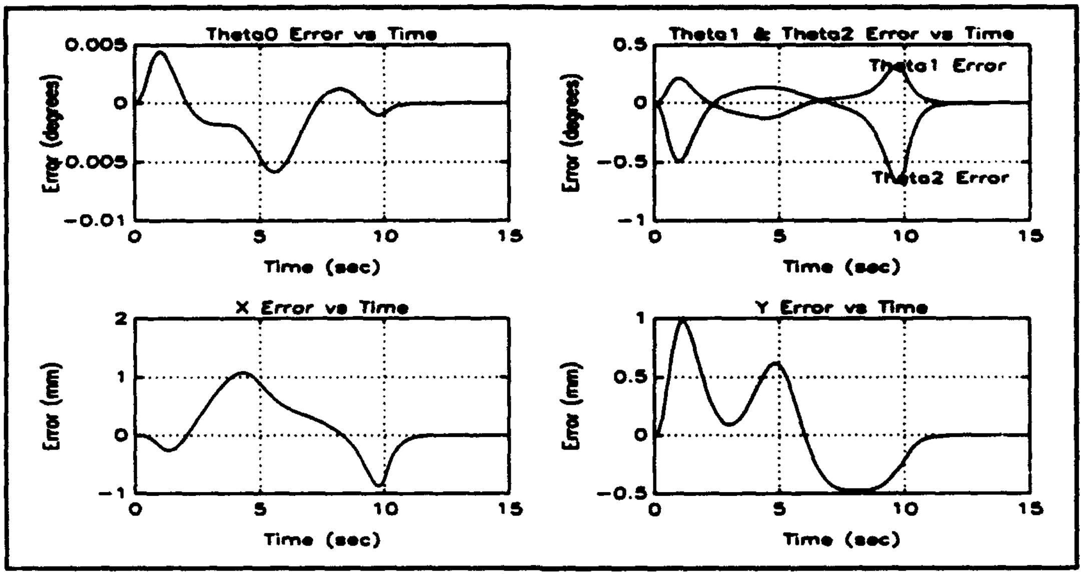

Figure 14: Linearizing Controller Error (100% parameter uncertainty)

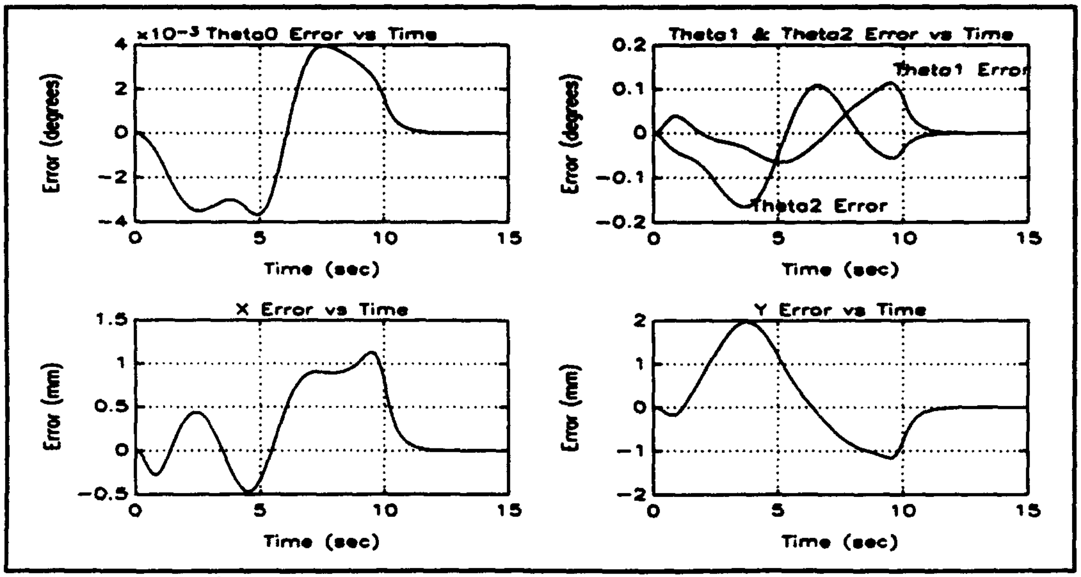

Figure 15: Reference Controller Error (100% parameter uncertainty)

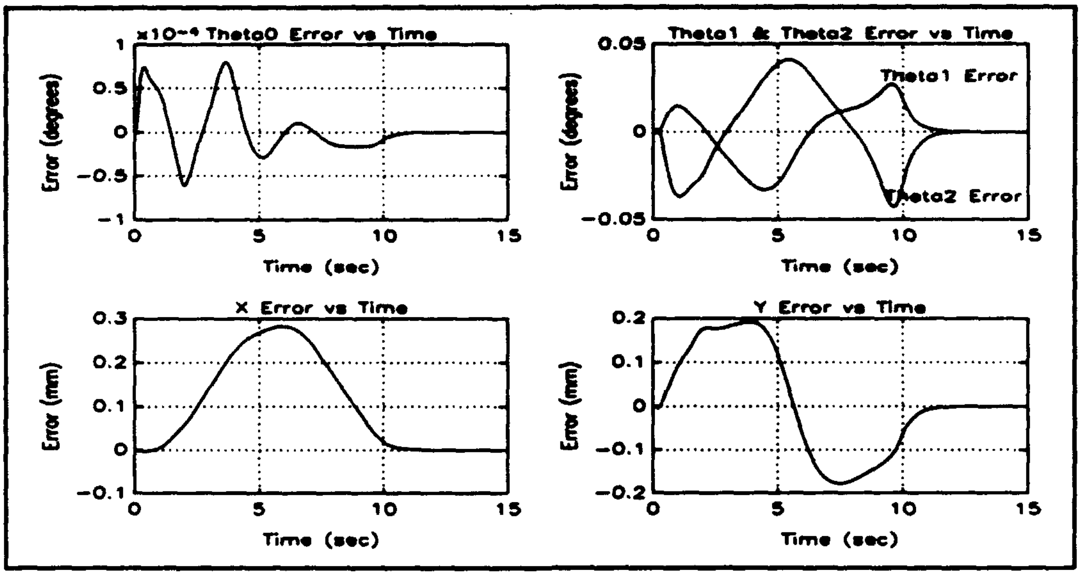

Figure 16: Adaptive Controller Error (100% parameter uncertainty)

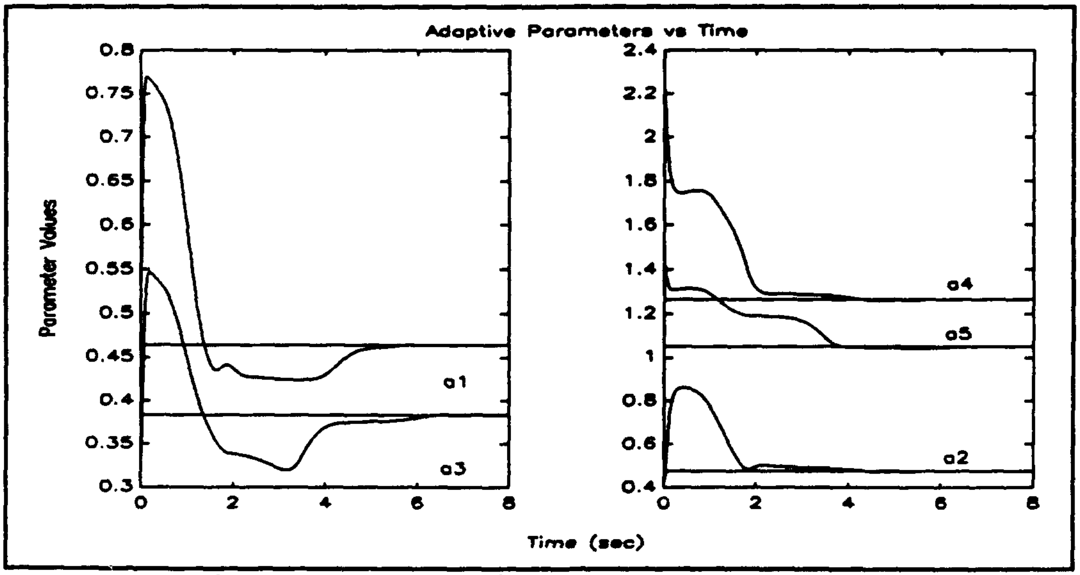

Figure 17: Adaptive Parameter Updates (100% parameter uncertainty)

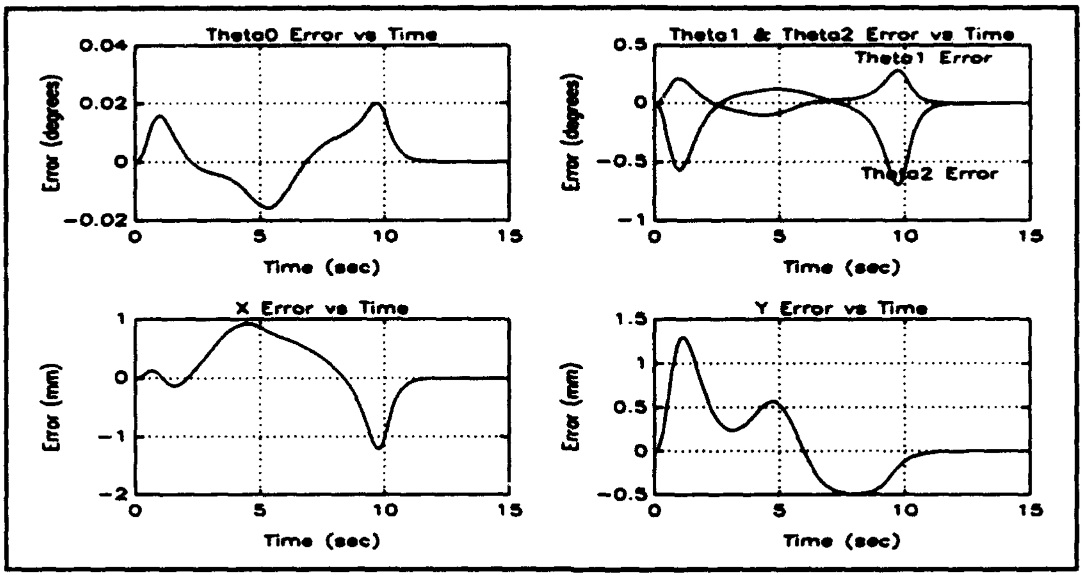

Figure 18: Linearizing Controller Error (150% parameter uncertainty)

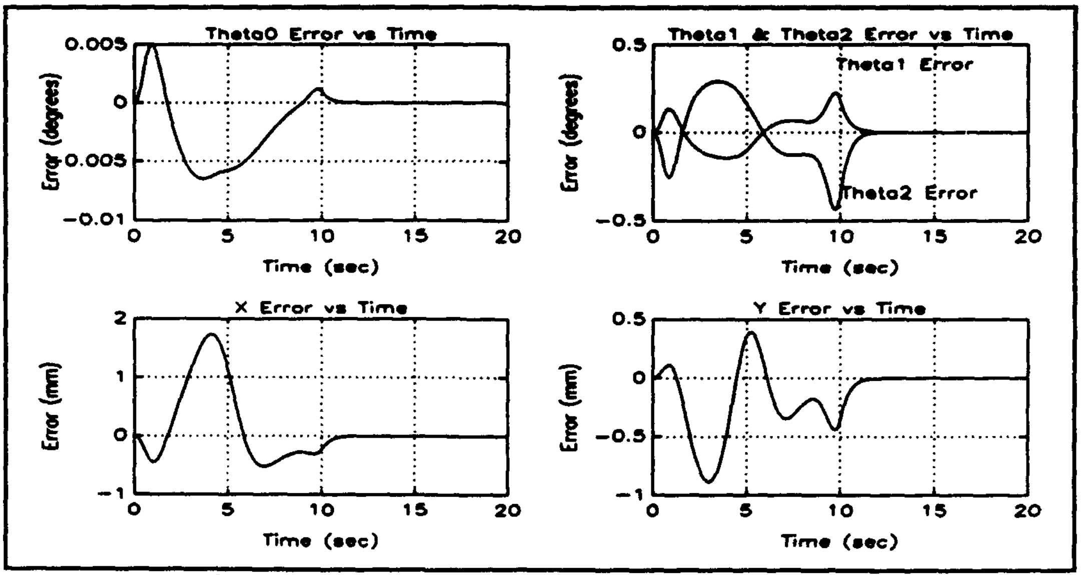

Figure 19: Reference Controller Error (150% parameter uncertainty)

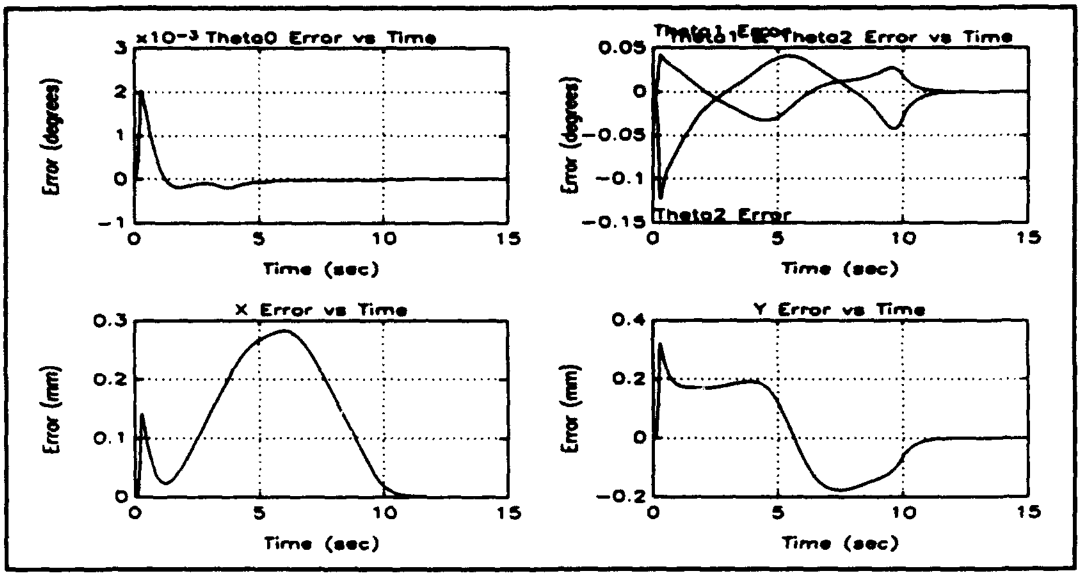

Figure 20: Adaptive Controller Error
(150% parameter uncertainty)

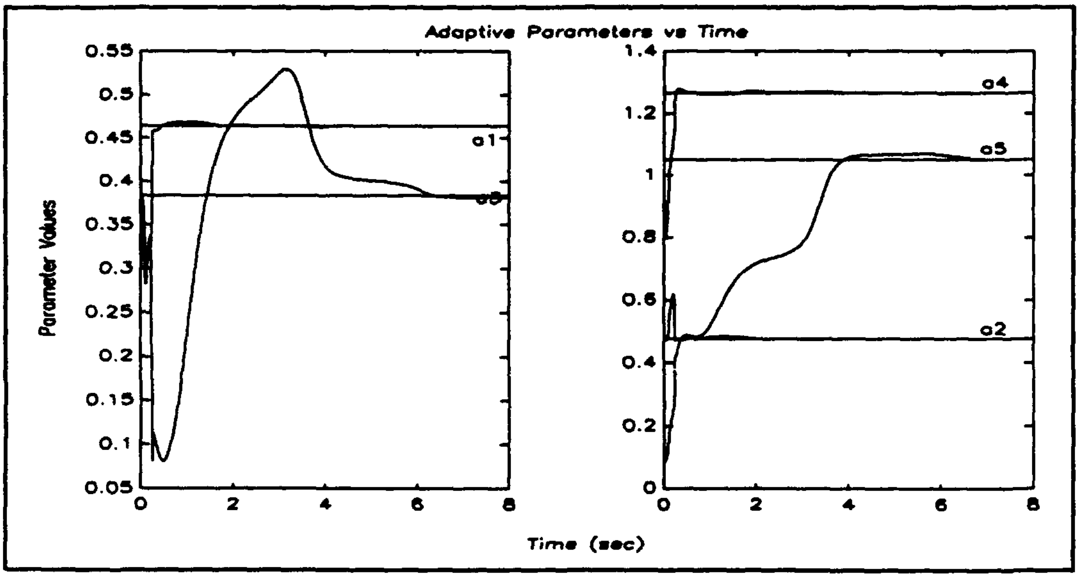

Figure 21: Adaptive Parameter Updates
(150% parameter uncertainty)

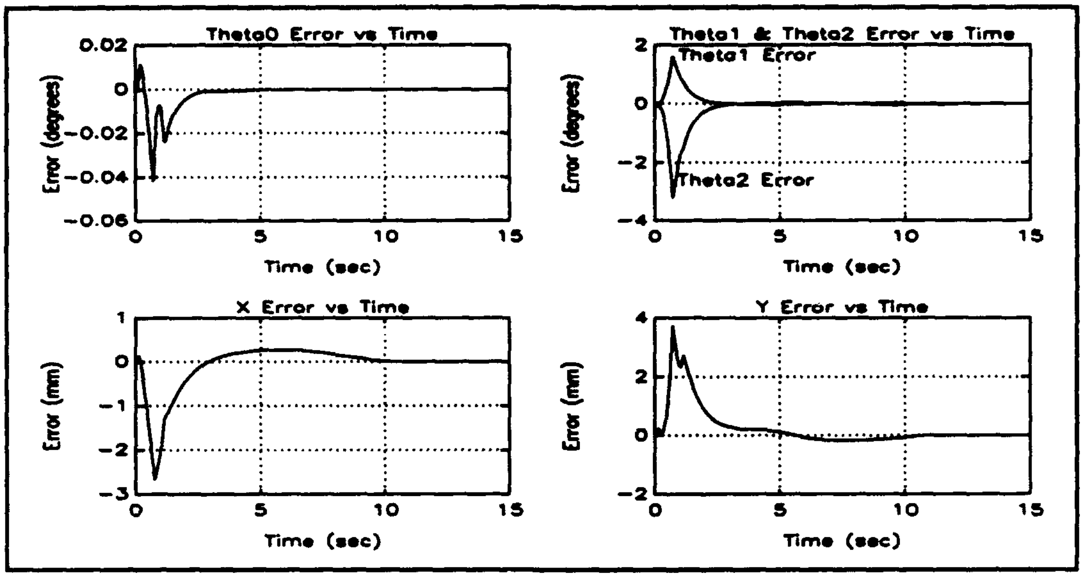

Figure 22: Adaptive Controller Error (500% parameter uncertainty)

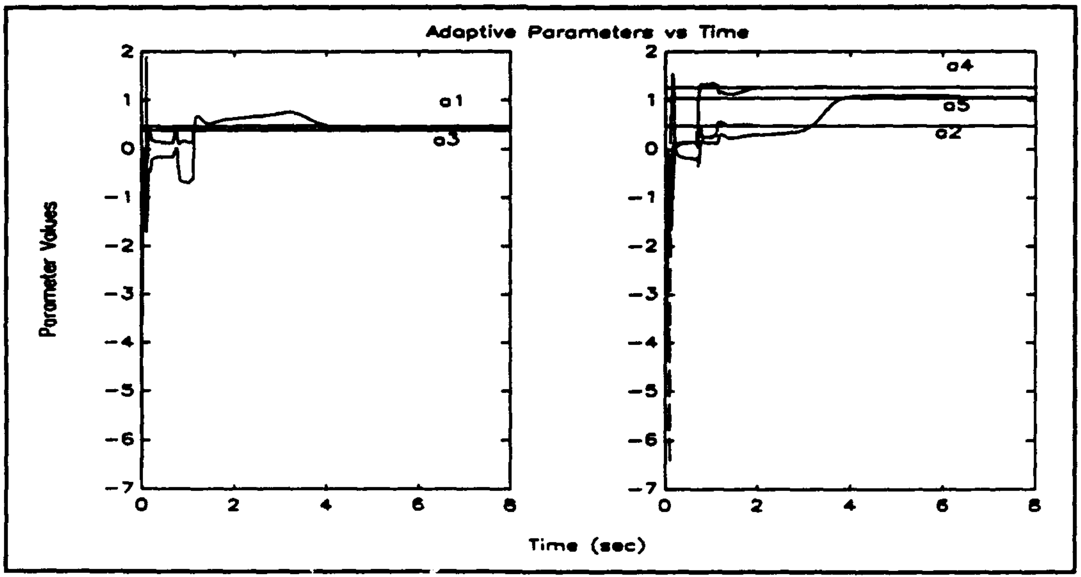

Figure 23: Adaptive Parameter Updates (500% parameter uncertainty)

IV. EXPERIMENTAL WORK

The spacecraft Robotics Simulator (SRS) was utilized for the experimental portion of the thesis. The SRS is a derivative of the Flexible Spacecraft Simulator (FSS) initially developed by Watkins [Ref. 10] and later modified by Hailey [Ref. 11]. Sorenson [Ref. 8] began the work to convert the FSS into the SRS. The robotic manipulator utilized was developed by Yale [Ref. 9].

A. SETUP

The SRS permits experimental investigation of two-dimensional robotics motion and rotational spacecraft dynamics and is illustrated in Figures 24 and 25. The simulation hardware is floated on an eight foot by six foot granite table by means of a thin layer of air supplied by an external source. The table is polished to within 0.001 inch peak to valley and leveled to prevent gravitational accelerations from impacting the motion across its surface. The following sections describe the simulated spacecraft with its associated sensors and actuators and the controller which together form the SRS. The spacecraft components are the centerbody and two-link robotic manipulator.

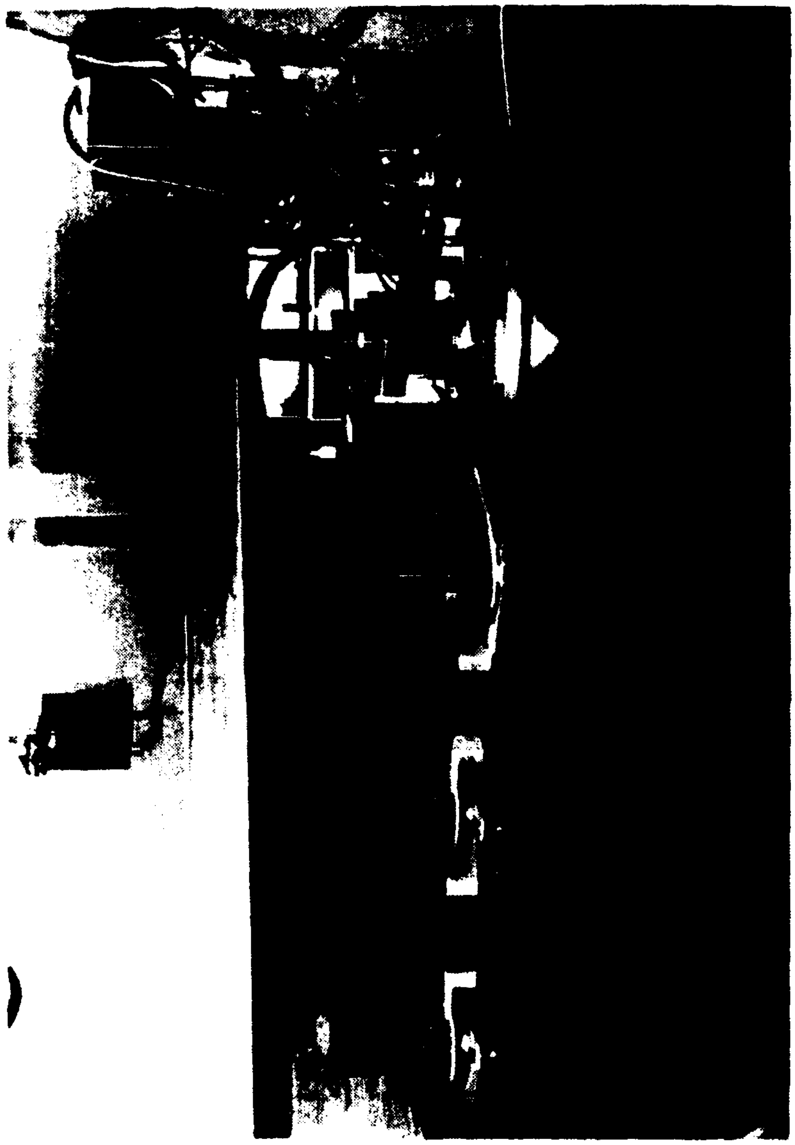

Figure 24: Spacecraft Robotic Simulator

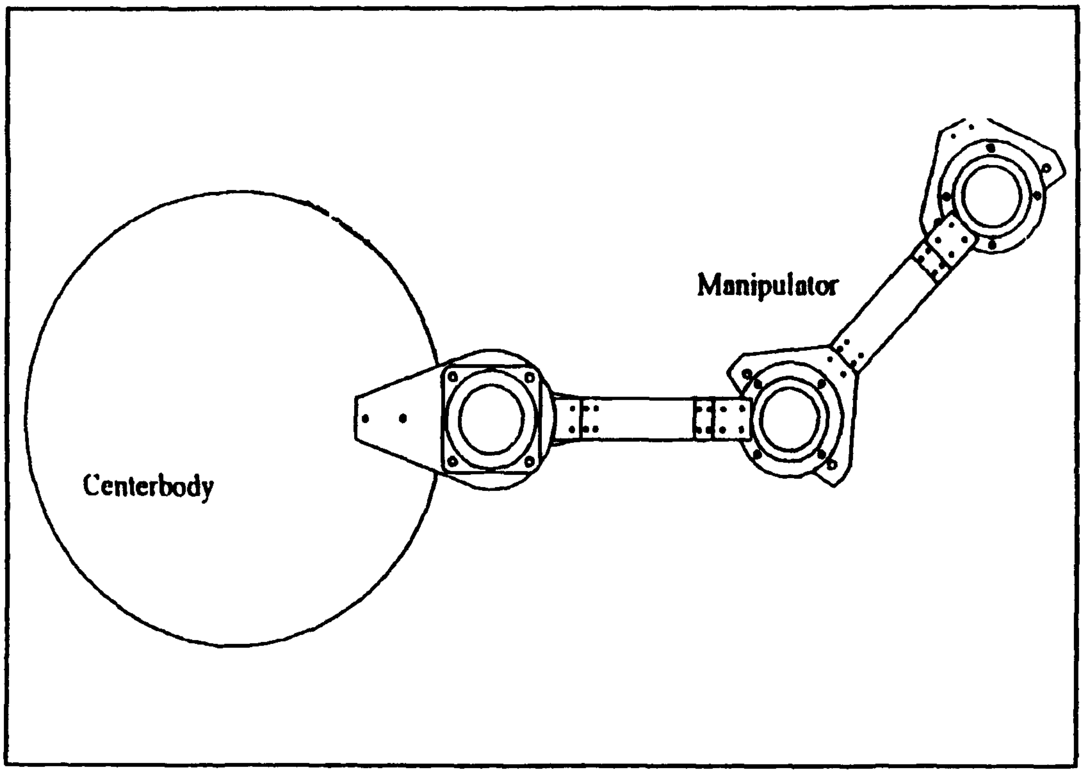

Figure 25: SRS Top View

1. Centerbody

The centerbody is a 30 inch diameter, 7/8 inch thick aluminum disk. It carries a position sensor, rate sensor, momentum wheel, thrusters, and an air tank to power the thrusters. The centerbody also serves as the mounting platform for the manipulators and is floated by a control air bearing and three air pads located at 120 degree intervals near the outer edge. The air pads are each capable of floating 60 pounds when the air pressure supplied to the pads is 80 psi. The centerbody is allowed to float freely on the granite table.

The centerbody angular rate is measured by a rate transducer manufactured by Humphrey, Inc., having a range of ±100 deg/sec and a minimum threshold of 0.01 deg/sec. The centerbody's translation is not measured and is neglected for the experiment.

The centerbody's angular position is controlled by a momentum wheel and is powered by a model JR16M4CH/F9T servo motor manufactured by PMI Industries whose characteristics are summarized in Table 3. Although the centerbody also carries two thrusters, they are not used in this research.

TABLE 3: MOMENTUM WHEEL MOTOR CHARACTERISTICS

Manufacturer	PMI Industries
Model	JR16M4CH/F9T
Rated Output Speed (rpm)	3000
Rated Current (amps)	7.79
Rated Voltage (volts)	168
Rated Torque (in-lb)	31.85
Height (in)	4.5
Weight (lb)	175
Outside Diameter (in)	7.4

2. Manipulators

The two-link manipulator has three joints. The shoulder joint is supported by the centerbody while the elbow and wrist joints are supported by two air pads apiece. The links between the joints are stiff laterally but permit some flexibility vertically. This feature increases the tolerances on the air pad height adjustment. Joint angles for the shoulder and elbow are sensed by encoders purchased with the joint actuators. The encoder resolution is 0.005 degrees. Manipulator actuators are harmonic drive dc servo actuators manufactured by HD Systems, Inc. The shoulder actuator is model RFS-25-6018-E036AL while the elbow and wrist actuators are model RFS-20-6012-E036AL. Specifications for joint actuators are contained in Table 4. The wrist joint actuator and sensor is not utilized in this experiment. Manipulator schematics are contained in Figure 26.

The joint actuators are all driven by Kepco power supplies. These bipolar, programmable, linear amplifiers can be controlled manually from the front panel or controlled remotely with a ±10 volt signal. In this application, The power supplies are operated in the current control mode with the voltage and current limits manually set consistent with the values in Table 4. The specific power supply models and their characteristics are summarized in Table 5.

TABLE 4: MANIPULATOR ACTUATOR CHARACTERISTICS

Manufacturer	HD Systems	HD Systems
Model	RFS-25-6012	RFS-25-6018
Reduction Ratio	1:50	1:50
Rated Output Speed (rpm)	60	60
Rated Current (amps)	2.9	3.9
Rated Voltage (volts)	75	75
Rated Torque (in-lb)	174	260
Height (in)	8.8	9.6
Weight (lb)	9.3	14.1
Footprint (in)	4.3 x 4.3	5.1 x 5.1

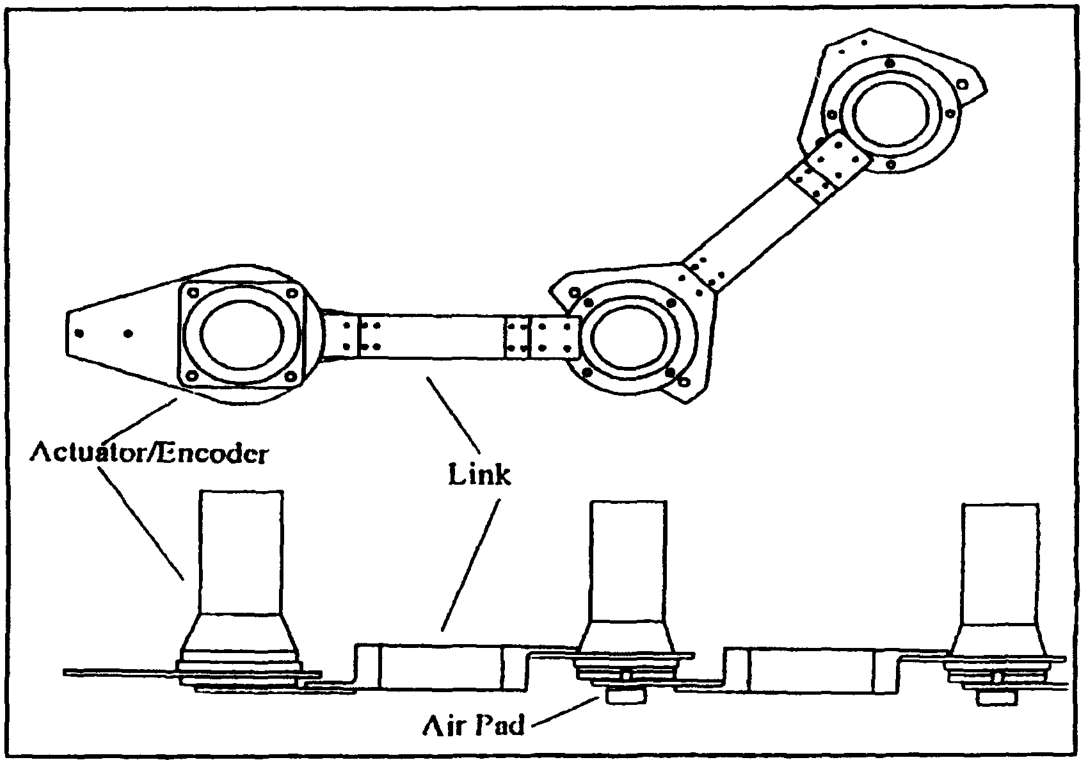

Figure 26: Manipulator Top and Side Views

3. Controller

The AC-100 programmable controller manufactured by Integrated Systems, Inc. controls the SRS. The AC-100 includes an Intel 80386 Application Processor, an Intel 80386 Multibus II Input/Output Processor, an Intel 80386 Communication Processor, an Intel 80387 Coprocessor, a Weitek 3167 Coprocessor, an Analog-To-Digital and Digital-To-Analog Data Translation DT2402 Input/Output Board, two INX-04 Encoder and Digital-To-Analog Servo Boards, and an Ethernet Interface Module. The AC-100 also includes software installed on a VAX-3100 Series Model 30 workstation. The software permits design of a controller in block diagram graphical form and conversion of the diagram to C language programming code. The user is also able to design an interactive animation window to operate the controller. The AC-100 receives input signals from the sensors and the graphical user interface. AC-100 output signals go to the power supplies driving the actuators or to the graphical user interface for display.

TABLE 5: POWER SUPPLY CHARACTERISTICS

Model	BOP 72-6M	BOP 72-3M
Actuator Controlled	Right Shoulder	Right Elbow
DC Output Range	±72 volts ±6 amps	±72 volts ±3 amps
Closed Loop Gain	0.6 (amp/volt)	0.3 (amp/volt)

4. System Integration

Several problems were encountered while attempting to implement the non-adaptive reference controller experimentally. As a result several modifications to the experiment were made.

a. *Actuator Dead Zones*

The first problem was caused by the large dead zones present in the harmonic drive motors. Both the shoulder and wrist motors were designed to be operated in a high torque environment. The SRS components are relatively small and offer little resistance to motion. This resulted in reference torques for a reasonable maneuver which were entirely within the dead zone of system actuators. This caused system tracking errors to build up until the PD control term produced torques larger than the actuator dead zones.

b. *Centerbody Resistance*

A second problem involved a noticeable resistance to rotation by the centerbody. This is due in part to the inability of the central air bearing to function except under very low lateral loading. This lateral loading was eliminated by allowing the centerbody to float freely and ignoring translation of the centerbody. This decreased some centerbody resistance to rotation but some resistance was still detected due to the effects of external wiring necessary for centerbody components.

c. Momentum Wheel Control

A third problem involved the ability to safely control the centerbody reaction wheel. The AC-100 control software was found to be subject to periodic freeze ups in which controller outputs could not be reliably predicted. The momentum wheel could therefore not be safely utilized. The experimental modification was to perform runs in which the centerbody was held fixed to simulate perfect reaction wheel performance.

B. RESULTS

Results are presented in Figures 27-40 for four control cases.

1. Case 1: High Gain, Free Centerbody

This case utilizes a high gain controller to control a system in which centerbody motion is not constrained. The controller gains utilized are presented in Table 6.

2. Case 2: Low Gain Controller, Free Centerbody

In this case, a low gain controller is utilized to a control a system in which the centerbody motion is not constrained. The controller gains utilized for this case are presented in Table 6.

3. Case 3: High Gain Controller, Fixed Centerbody

The controller utilized in this case is identical to that of Case 1 but the centerbody is now held in a fixed position.

4. Case 4: Low Gain Controller, Fixed Centerbody

This controller is identical to Case 2 but is used to control a system in which the centerbody position is held fixed as in Case 3.

C. COMPARISON OF CONTROLLERS

1. High vs Low Gain

The high gain controllers yielded lower steady state errors than the low gain controllers at the cost of larger oscillations during the maneuver.

2. Free vs Fixed Centerbody

When the centerbody is allowed to float freely, significant errors in centerbody attitude and manipulator tip position are seen.

TABLE 6: CONTROLLER GAINS

	Shoulder Gain	Elbow Gain
K_p (low gain controller)	500	1000
K_p (high gain controller)	1000	2000
K_v (low gain controller)	10	25
K_v (high gain controller)	20	50

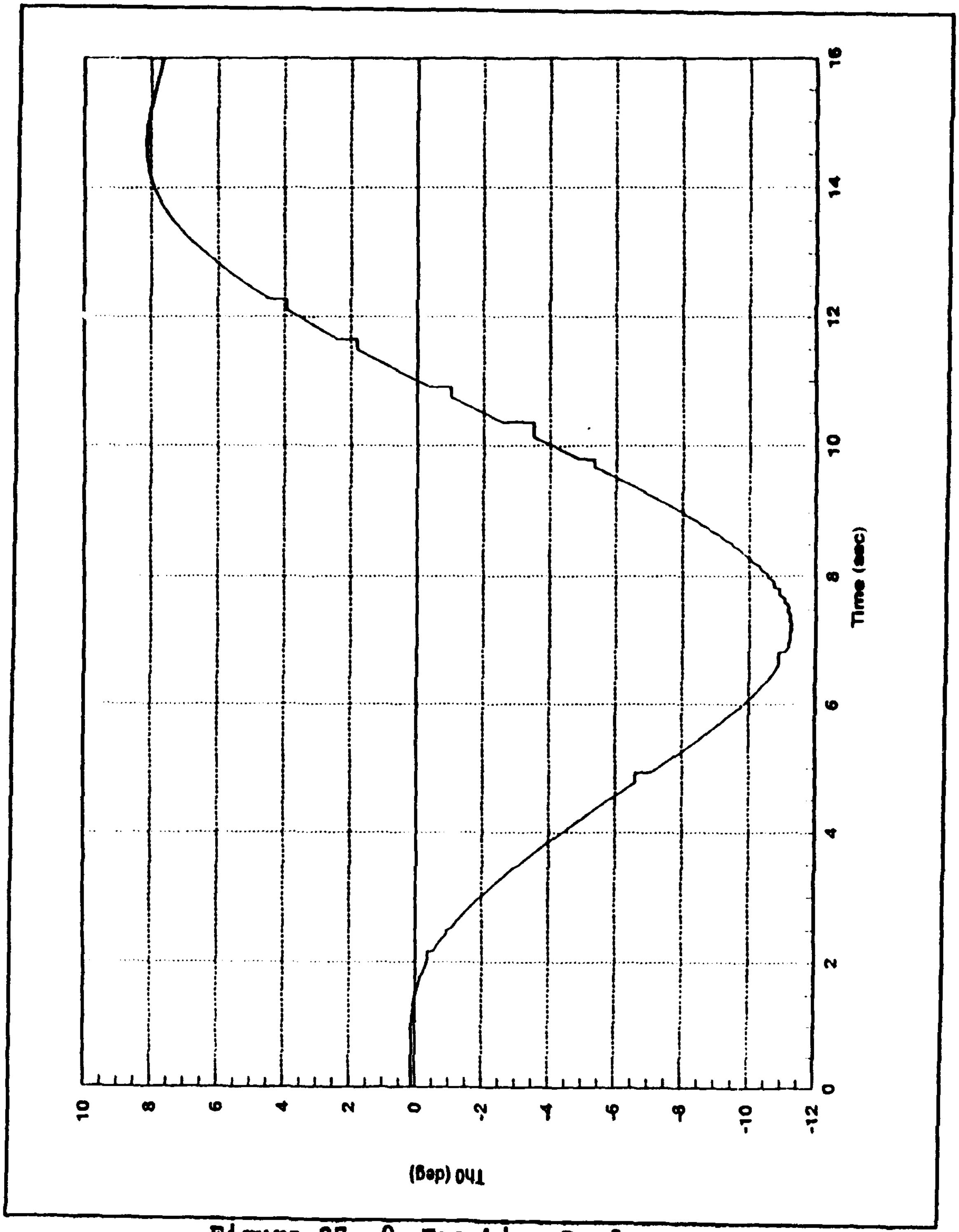

Figure 27: θ_0 Tracking Performance (High Gain, Free Centerbody)

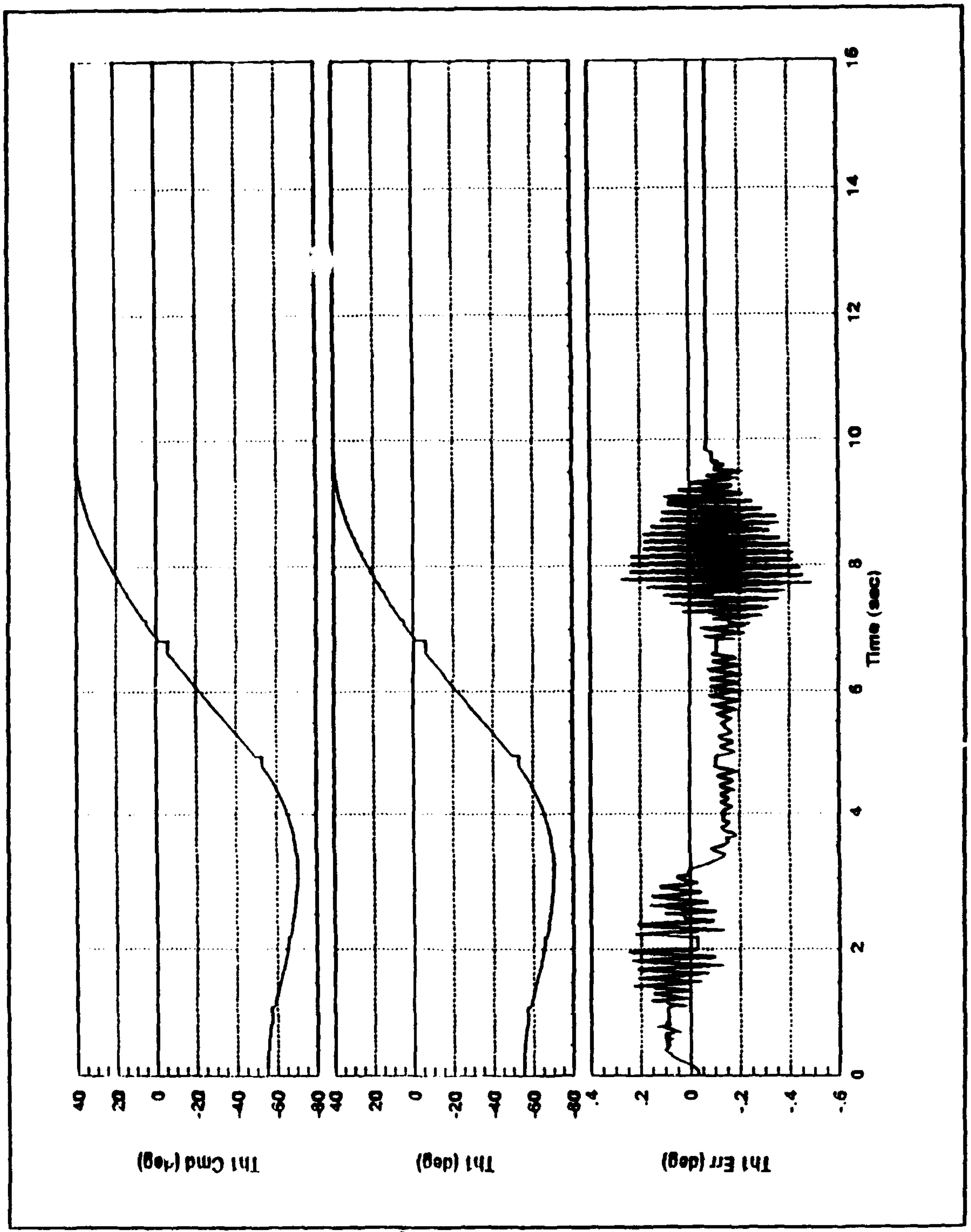

Figure 28: θ_1 Tracking Performance
(High Gain, Free Centerbody)

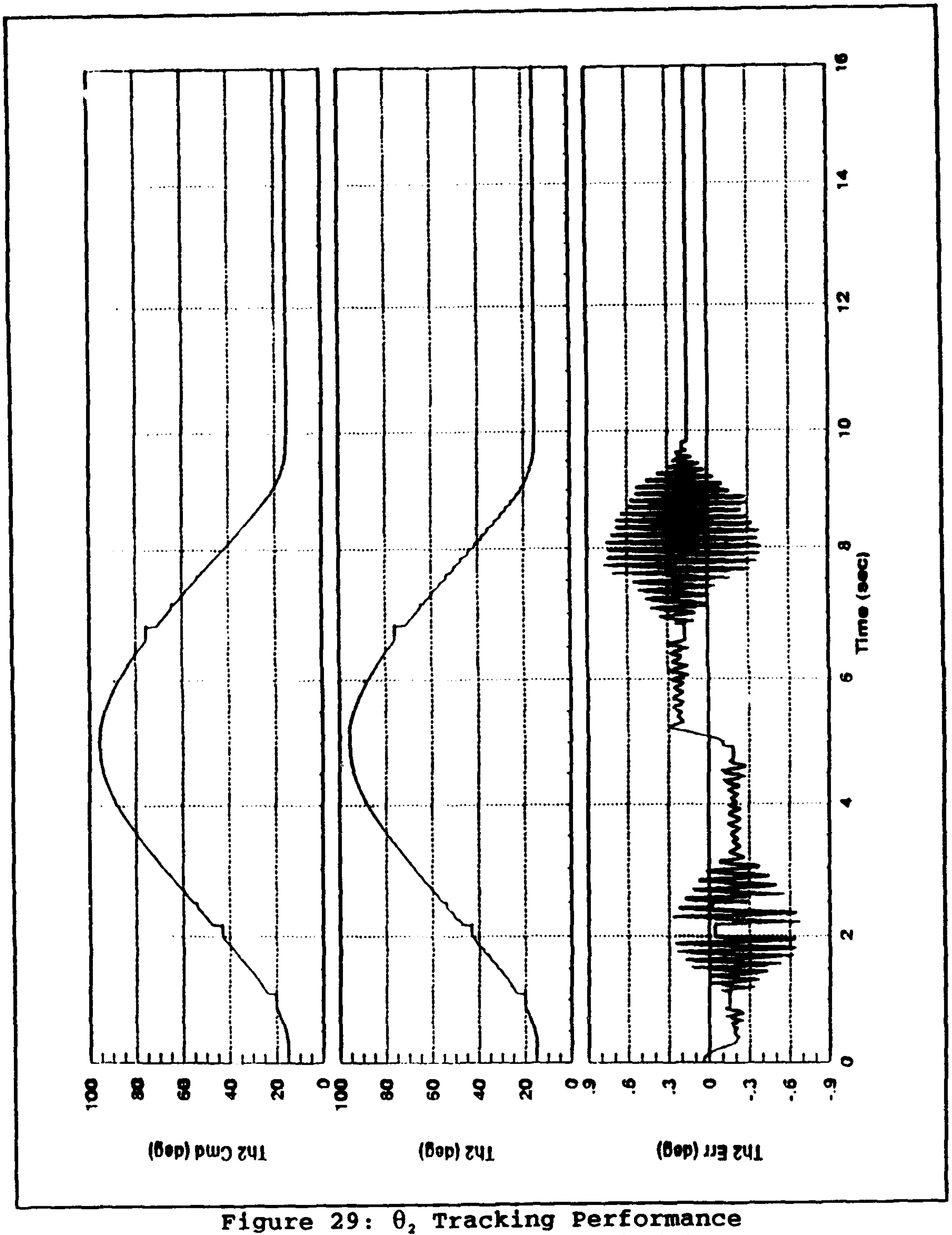

Figure 29: θ_2 Tracking Performance
(High Gain, Free Centerbody)

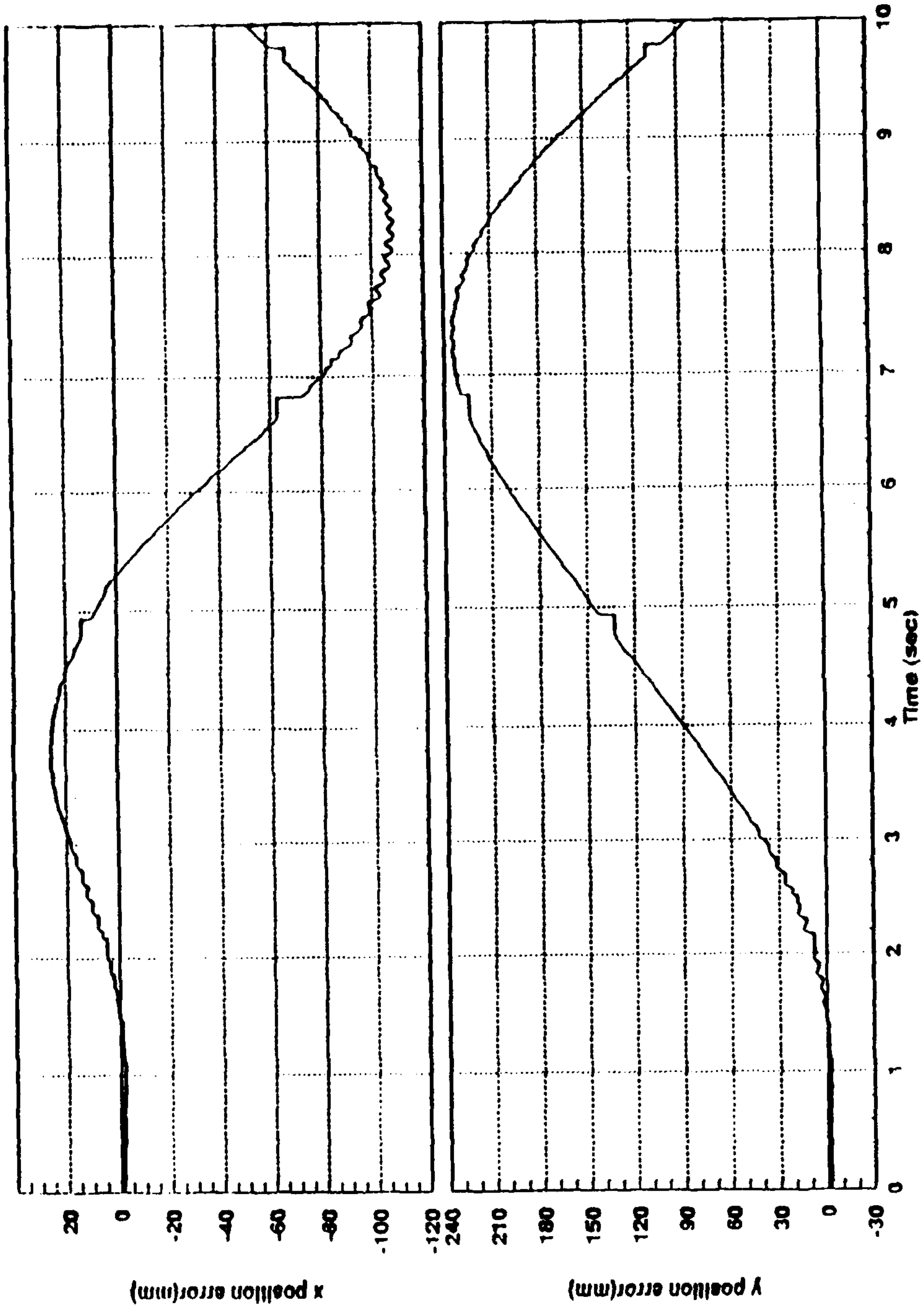

Figure 30: Tip Tracking Performance
(High Gain, Free Centerbody)

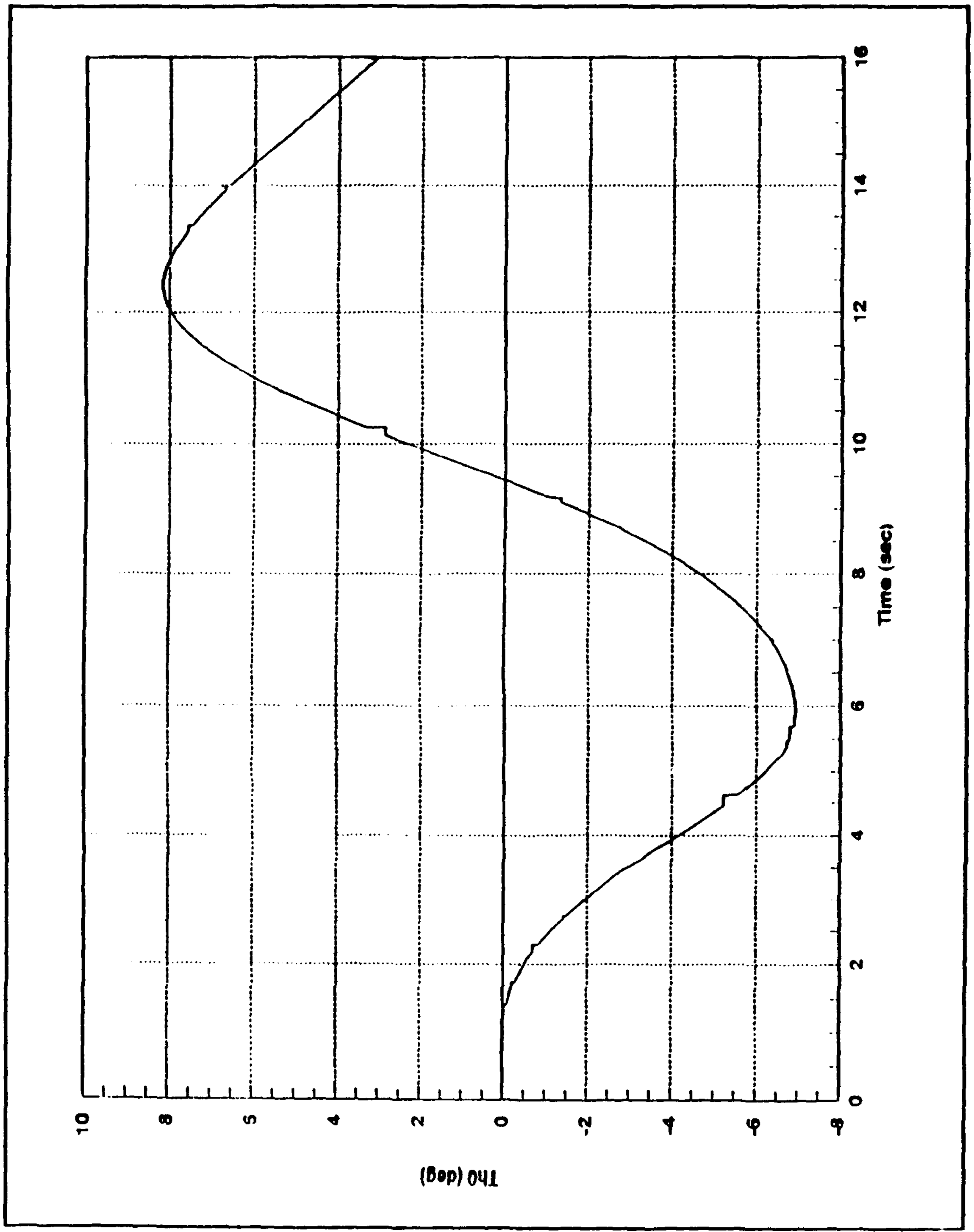

Figure 31: θ_0 Tracking Performance
(Low Gain, Free Centerbody)

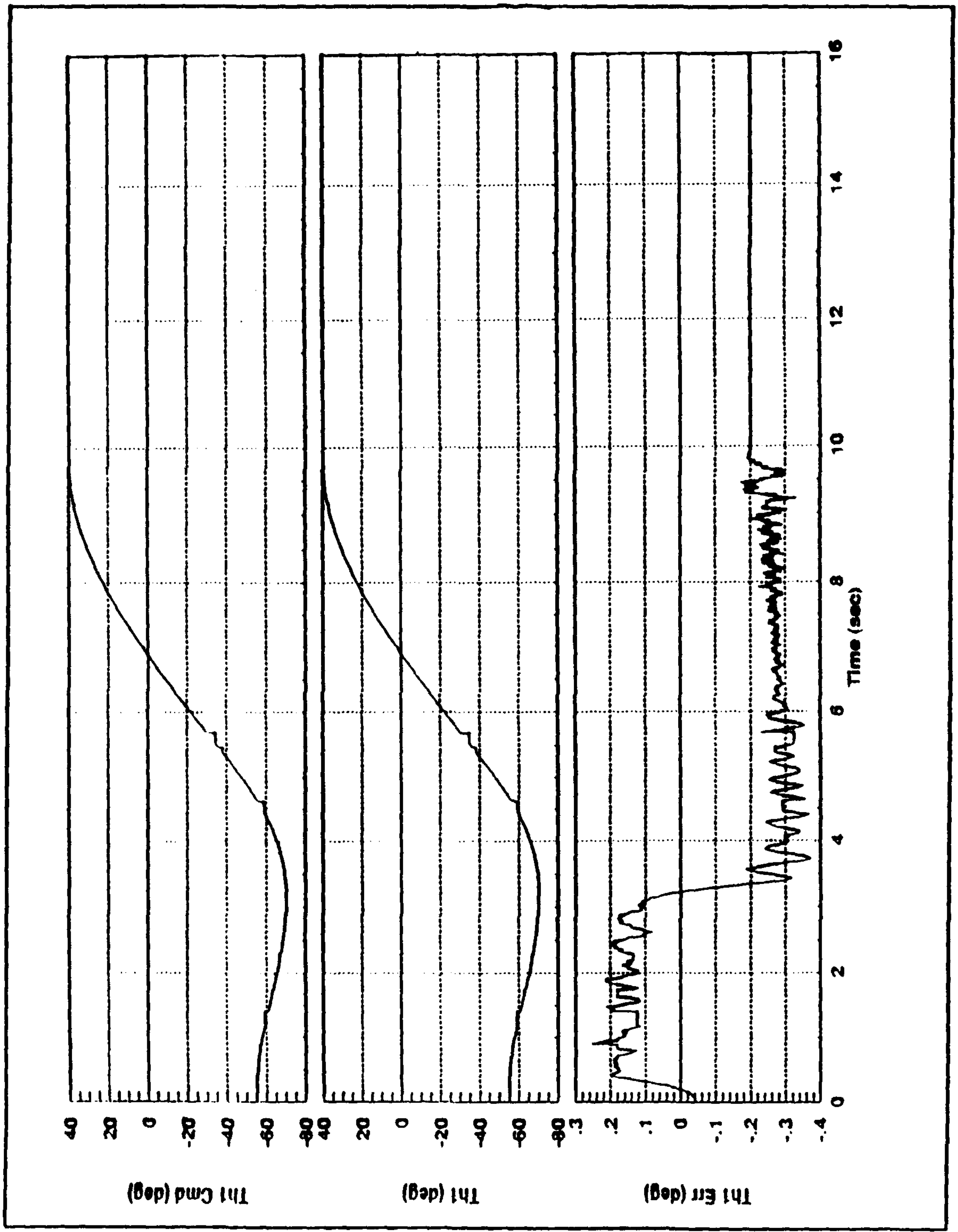

Figure 32: θ_1 Tracking Performance
(Low Gain, Free Centerbody)

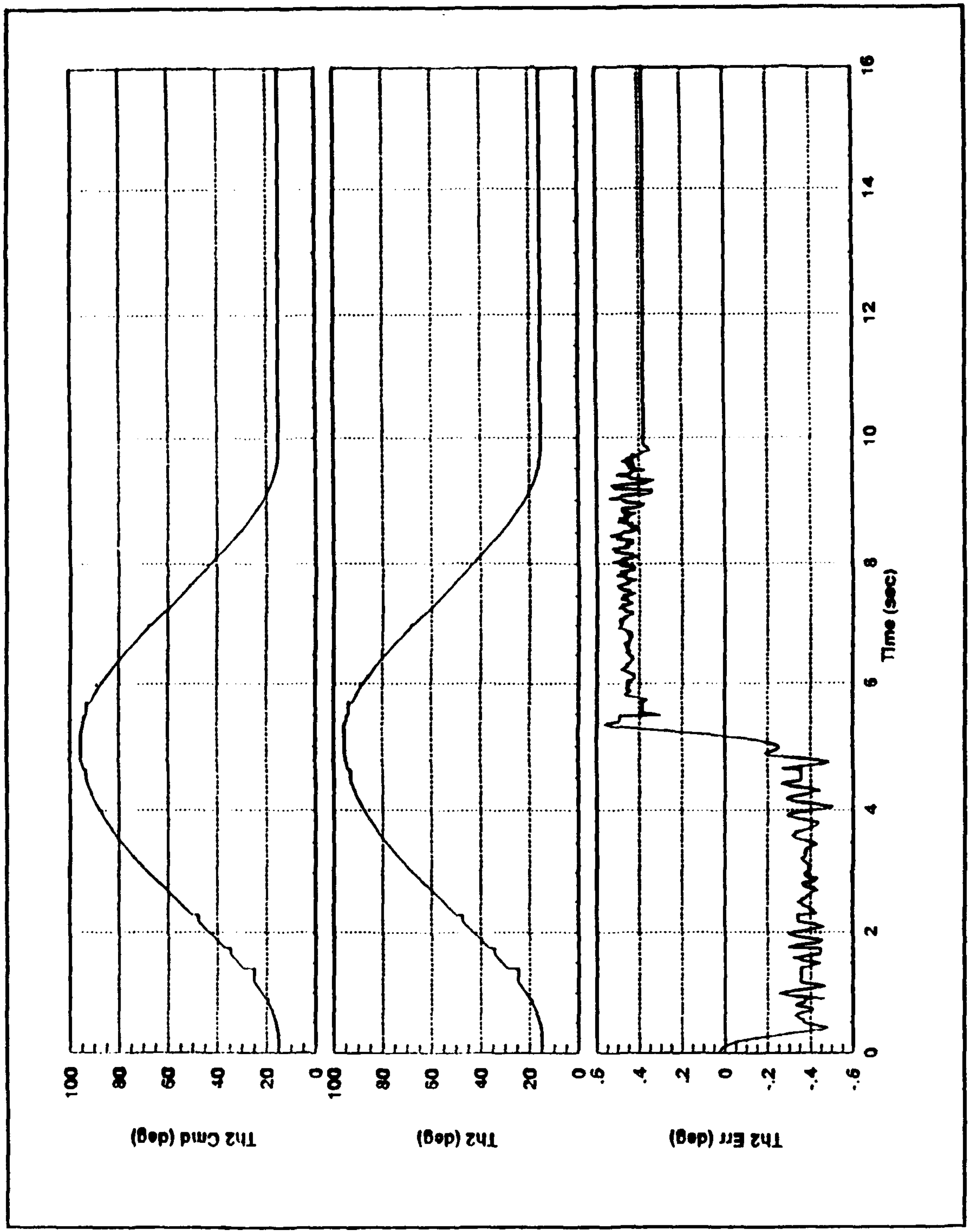

Figure 33: θ_2 Tracking Performance
(Low Gain, Free Centerbody)

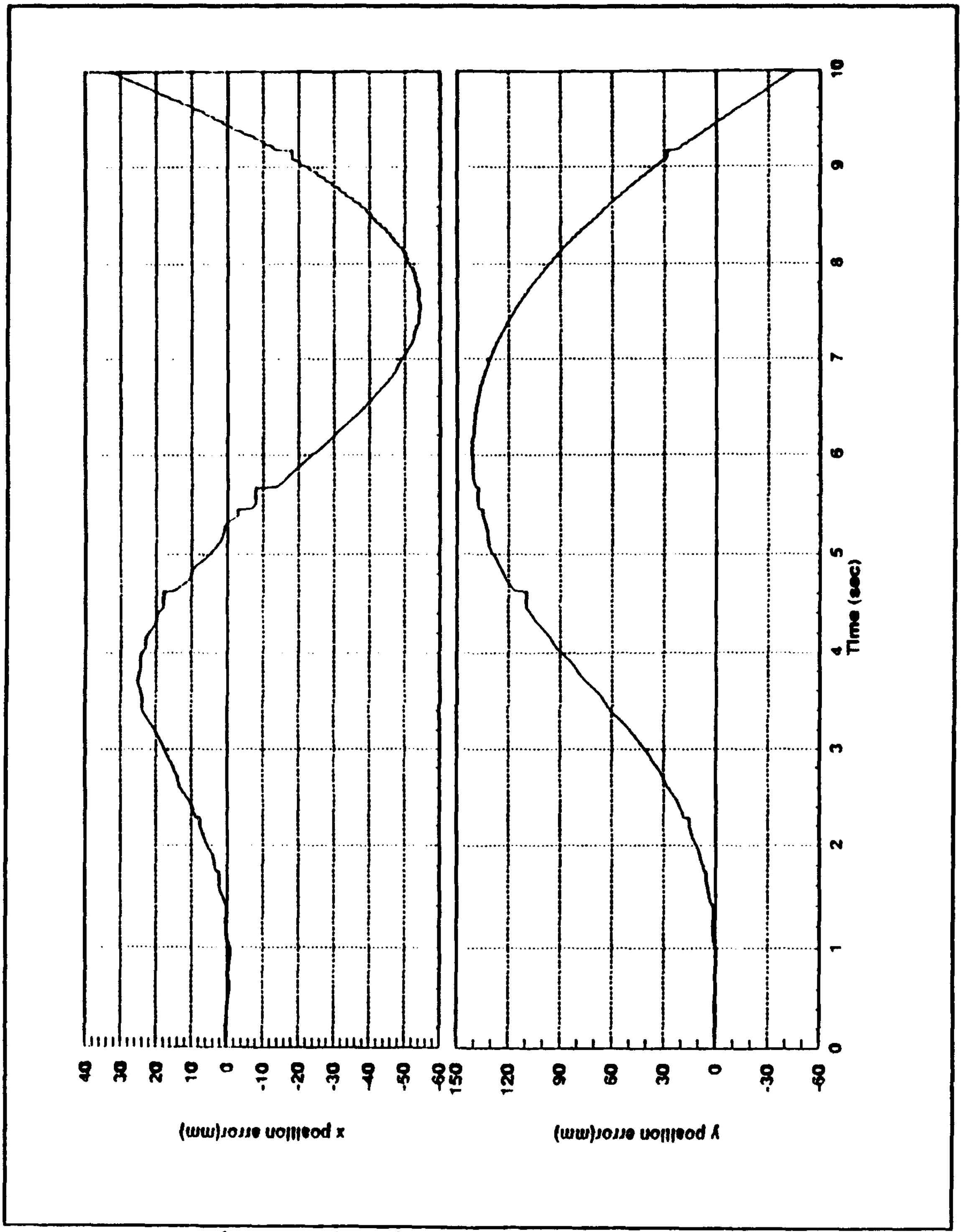

Figure 34: Tip Tracking Performance
(Low Gain, Free Centerbody)

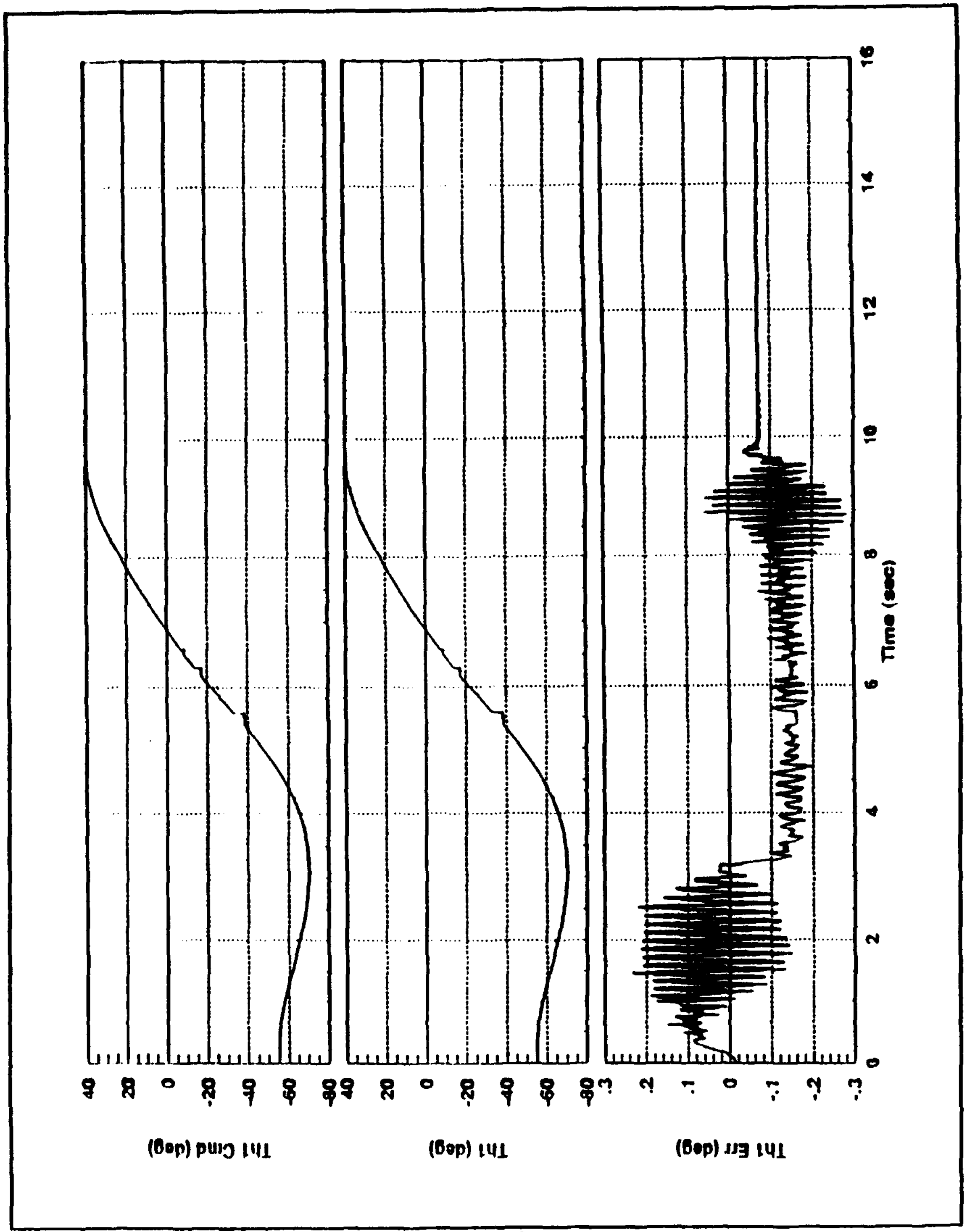

Figure 35: θ_1 Tracking Performance
(High Gain, Fixed Centerbody)

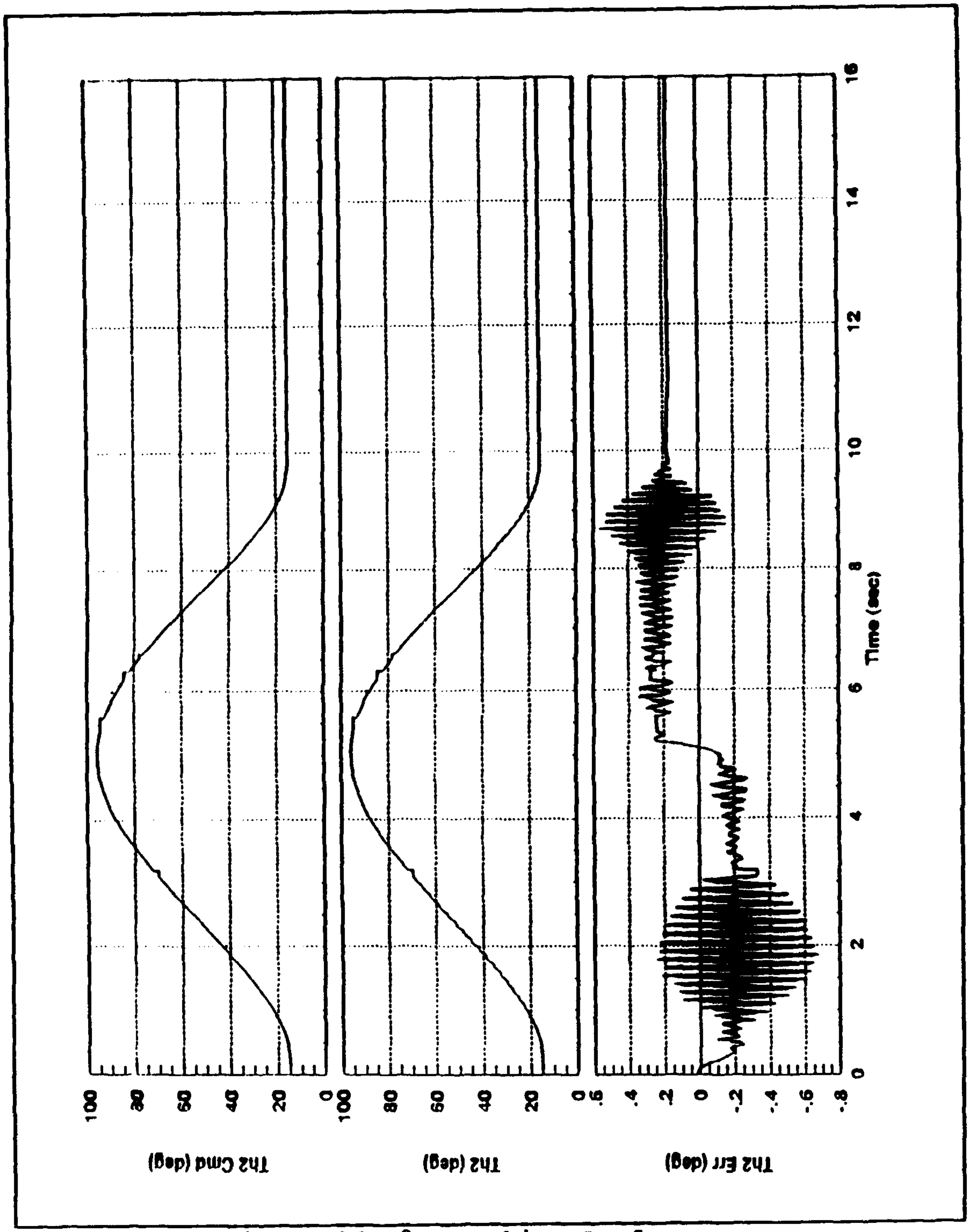

Figure 36: θ_2 Tracking Performance
(High Gain, Fixed Centerbody)

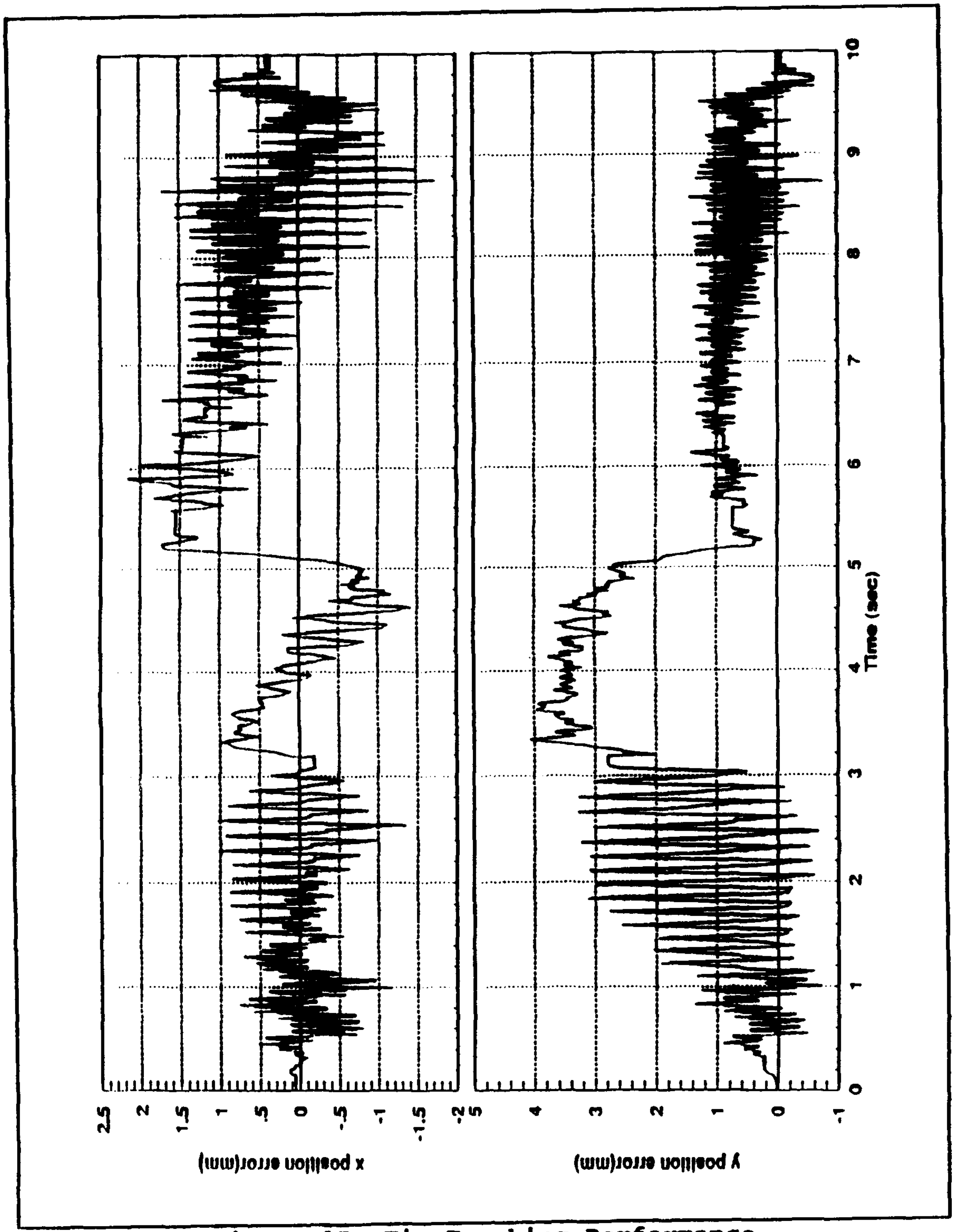

Figure 37: Tip Tracking Performance
(High Gain, Fixed Centerbody)

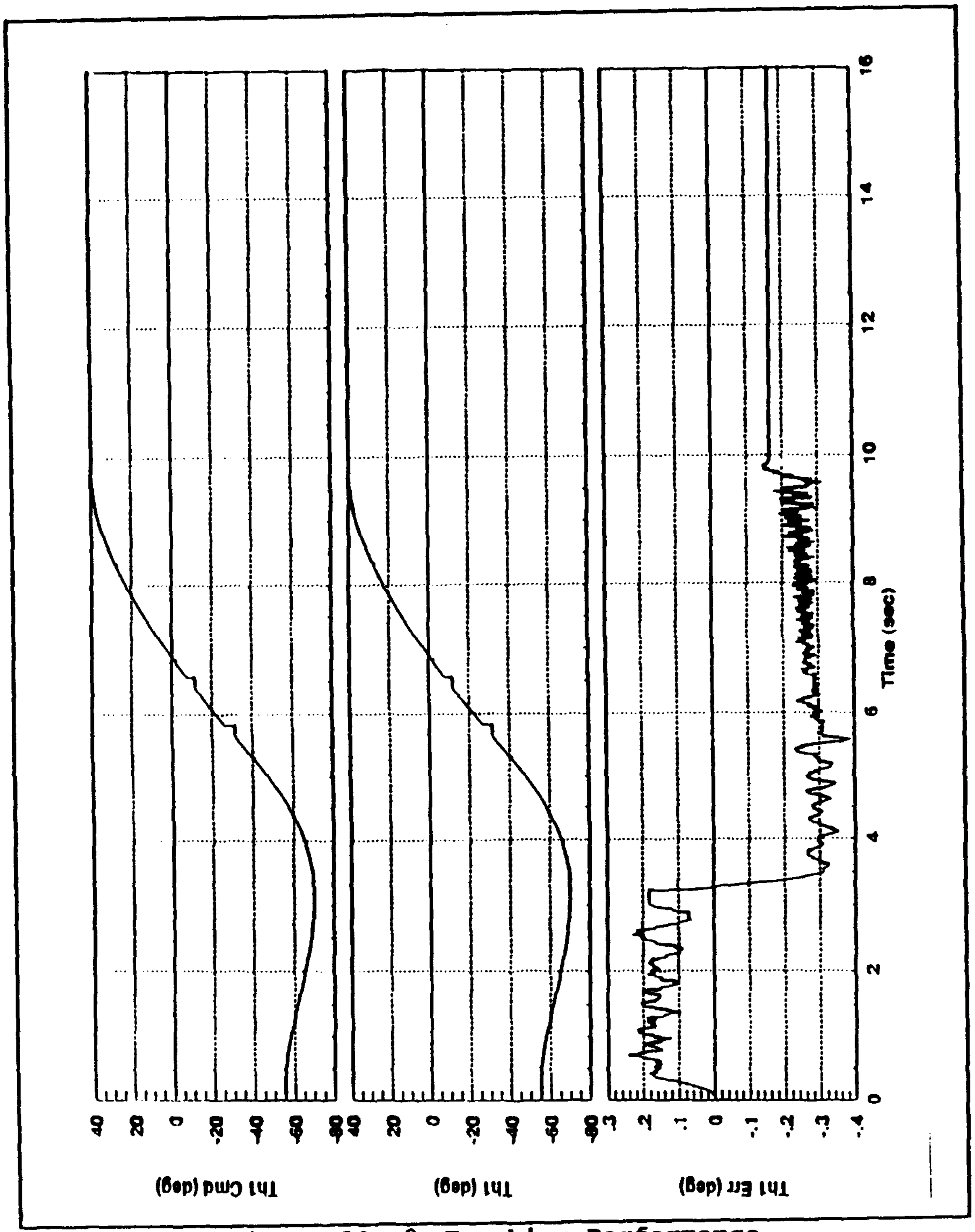

Figure 38: θ_1 Tracking Performance
(Low Gain, Fixed Centerbody)

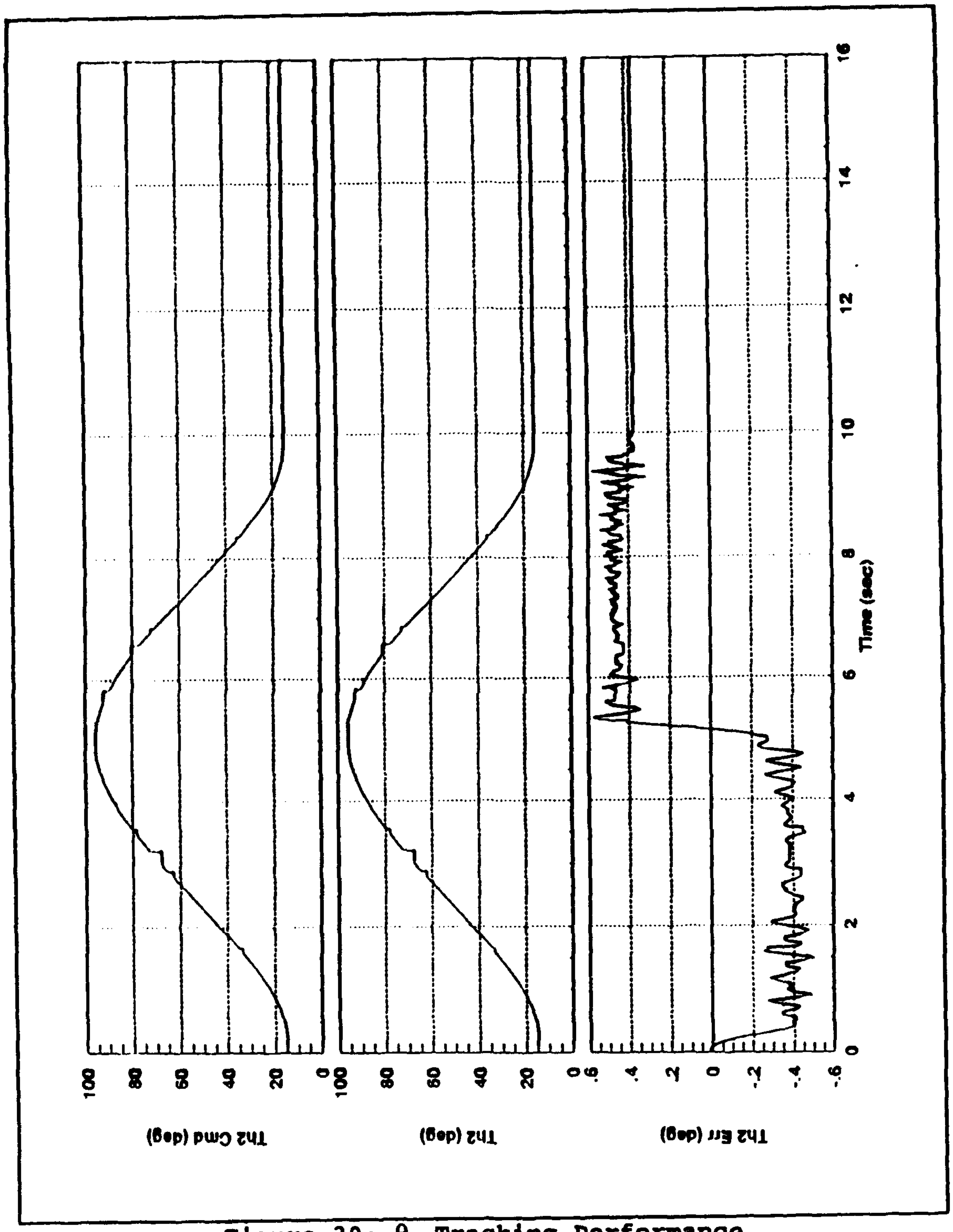

Figure 39: θ_2 Tracking Performance (Low Gain, Fixed Centerbody)

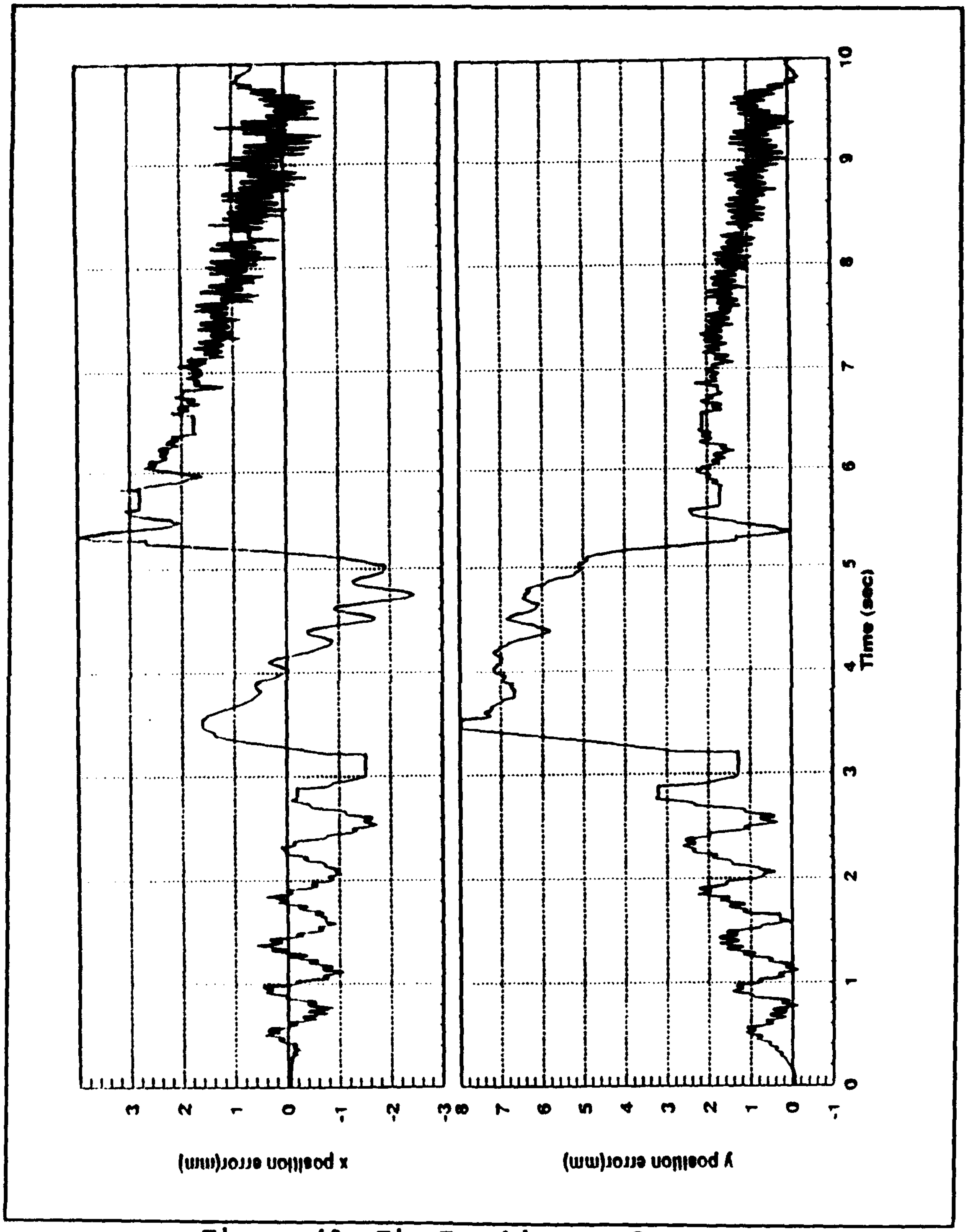

Figure 40: Tip Tracking Performance
(Low Gain, Fixed Centerbody)

V. SUMMARY AND CONCLUSIONS

A. SUMMARY

Three controllers were successfully developed for a spaced based two-link robotic manipulator. Two non-adaptive controllers (a linearizing controller and a reference controller) were first developed using feedback linearization techniques. An adaptive controller was then developed through the linear parameterization of system dynamics and the use of a recursive Kalman Filter based adaption law. Controllers were then compared for system parameter errors up to ±500%.

Centerbody pointing accuracy was improved by utilizing adaptive control while centerbody control torque was effected very little. For high values of parameter uncertainty, manipulator tracking errors were smaller when using the adaptive controller. For low values of parameter uncertainty, the linearizing non-adaptive controller outperformed the adaptive controller in some areas.

Implementation of the non-adaptive reference controller experimentally demonstrated the effects of un-modelled system dynamics. The oscillations and steady state errors encountered only reinforced the value of adaptive control in real world applications. High PD control gains produced larger oscillations but smaller steady state errors than low gains.

Manipulator maneuvers produced significant disturbances on centerbody attitude when no control was applied to hold the centerbody steady.

B. RECOMMENDATIONS FOR FURTHER STUDY

Feedback linearization represents only one way to attack a non-linear control problem. Other approaches include neural networks and sliding mode controllers.

The adaptive controller developed may be too cumbersome to implement in real time. In order to decrease computation time the system can be re-parameterized in terms of only payload characteristics and a more efficient adaption law developed.

Experimental implementation inaccuracies resulted from un-modelled actuator dead zones. System equations of motion and reference torques can be reformulated to include this effect. The use of fuzzy logic is also worth investigating.

APPENDIX A: MEMBER KINETIC ENERGIES

Kinetic energy of individual components is found using Eq.(7)

$$T_i=\frac{1}{2}I_i\omega_i^2+\frac{1}{2}m_i(\dot{\boldsymbol{r}}\cdot\dot{\boldsymbol{r}}) \tag{7}$$

The centerbody angular rate and center of mass position vector are given by

$$\boldsymbol{\omega}_0=\boldsymbol{\theta}_0 \tag{79}$$

$$\dot{\boldsymbol{r}}_0=L_{c0}\hat{x}_0 \tag{80}$$

Differentiating Eq.(80) results in the velocity of the centerbody center of mass

$$\dot{\boldsymbol{r}}_0=L_{c0}\omega_0\hat{y}_0 \tag{81}$$

Substituting Eq.(79) and (81) back into Eq.(7) and collecting on the angular rate term leads to

$$T_0=\frac{1}{2}(I_0+m_0L_{c0}^2)\theta_0^2 \tag{82}$$

Similar developments are used for each of the remaining pieces in the system. For the manipulator link between the shoulder and elbow (Link 1), the angular velocity is a combination of centerbody rotation and rotation of the link with respect to the centerbody.

$$\omega_1=\theta_0+\theta_1 \qquad (83)$$

The position vector to the center of mass is

$$\underline{r}_1=L_0\cos\theta_0\hat{x}_0+L_0\sin\theta_0\hat{y}_0+L_{c1}\hat{x}_1 \qquad (84)$$

The first two terms in the position vector represent the location of the shculder. Differentiating the position vector gives the expression for the velocity vector. Because none of the coordinate axes used in the position vector expression are inertial, their rotation must be included as well.

$$\dot{\underline{r}}_1=L_0\cos\theta_0\hat{y}_0-L_0\omega_0\sin\theta_0\hat{x}_0+L_{c1}\omega_1\hat{y}_1 \qquad (85)$$

After Eqs.(83) and (84) are substituded into Eq.(7) and terms are grouped by angular rates, the kinetic energy is

$$\begin{aligned}T_1=&\theta_0^2(0.5(I_1+m_1L_{c1}^2+m_1L_0^2)+m_1L_0L_{c1}(\sin\theta_0\sin(\theta_0+\theta_1)+\cos(\theta_0+\theta_1)))\\&+0.5\theta_1^2(I_1+m_1L_{c1}^2)\\&+\dot{\theta}_0\theta_1(I_1+m_1L_{c1}^2+m_1L_0L_{c1}(\sin\theta_0\sin(\theta_0+\theta_1)+\cos\theta_0\cos(\theta_0+\theta_1)))\end{aligned} \qquad (86)$$

The angular rates for the link between the shoulder and wrist (Link 2) includes the centerbody angular rates as well as the angular rates of the body axes on Links 1 and 2.

$$\omega_2=\theta_0+\theta_1+\theta_2 \qquad (87)$$

This link's center of mass position vector is

$$\underline{r}_2=L_0\cos\theta_0\hat{x}_0+L_0\sin\theta_0\hat{y}_0+L_1\hat{x}_1+L_{c2}\hat{y}_2 \qquad (88)$$

Differentiating the position vector gives the velocity vector

$$\dot{L}_2 = L_0\omega_0\cos\theta_0\hat{y}_0 - L_0\omega_0\sin\theta_0\hat{x}_0 + L_1\omega_1\hat{y}_1 + L_{c2}\omega_2\hat{y}_1 \qquad (89)$$

The kinetic energy for link 2 is found after substituting Eqs.(87) and (89) into Eq.(6) and collecting terms with common angular rates.

$$\begin{aligned}
T_2 = \dot{\theta}_0^2 (0.5 (& I_2 + m_2L_{c2}^2 + m_2L_1^2 + m_2L_0^2) \\
& + m_2L_0L_1(\sin\theta_0\sin(\theta_0+\theta_1) + \cos\theta_0\cos(\theta_0+\theta_1) + m_2L_1L_{c2}\cos\theta_2 \\
& + m_2L_0L_{c2}(\sin\theta_0\sin(\theta_0+\theta_1+\theta_2) + \cos\theta_0\cos(\theta_0+\theta_1+\theta_2))) \\
+ \dot{\theta}_1^2 & (0.5 (I_2 + m_2L_1^2 + m_2L_{c2}^2) + m_2L_1L_{c2}\cos\theta_2) + 0.5\dot{\theta}_2^2 (I_2 + m_2L_{c2}^2) \\
+ \dot{\theta}_0\dot{\theta}_1 & (I_2 + m_2L_1^2 + m_2L_{c2}^2 + 2m_2L_1L_{c2}\cos\theta_2 \\
& + m_2L_0L_1(\sin\theta_0\sin(\theta_0+\theta_1) + \cos\theta_0\cos(\theta_0+\theta_1)) \\
& + m_2L_0L_{c2}(\sin\theta_0\sin(\theta_0+\theta_1+\theta_2) + \cos\theta_0\cos(\dot{\theta}_0+\theta_1+\theta_2))) \\
+ \dot{\theta}_0\dot{\theta}_2 & (I_2 + m_2L_{c2}^2 + m_2L_1L_{c2}\cos\theta_2 \\
& + m_2L_0L_{c2}(\sin\theta_0\sin(\dot{\theta}_0+\theta_1+\theta_2) + \cos\theta_0\cos(\theta_0+\theta_1+\theta_2))) \\
+ \dot{\theta}_1\dot{\theta}_2 & (I_2 + m_2L_{c2}^2 + m_2L_1l_{c2}\cos\theta_2)
\end{aligned}$$

APPENDIX B: MATLAB CODE

A. PCONT

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Main Program to Track a Polynomial Reference Maneuver %
% pcont.m                                               %
% calls: ode2.m                                         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clg
clear

%%%%%%%%%%%%%%%%%
% Define States %
%%%%%%%%%%%%%%%%%

% x(1)=thd1
% x(2)=thd2
% x(3)=thd3
% x(4)=th1
% x(5)=th2
% x(6)=th3


%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Actual System Constants %
%%%%%%%%%%%%%%%%%%%%%%%%%%%

L0 = .427;    % centerbody radius
L1 = .530;    % length of arm 1
L2 = .533;    % length of arm 2

Lc0 = .104;         % length to CM of centerbody
Lc1 = .403;         % length to CM arm 1
Lc2 = .314;         % length to CM arm 2

m0 = 65.96;          % centerbody mass
m1 = 2.34;           % arm 1 mass
m2 = 2.86;           % arm 2 mass

I0 = 5.74;        % centerbody inertia
I1 = .081;        % arm 1 inertia
I2 = .182;        % arm to inertia

%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Actual System Parameters %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%

aa(1)= I2 + m2*Lc2^2;
aa(2)= m2*L1*Lc2;
aa(3)= m2*L0*Lc2;
aa(4)= I1 + m1*Lc1^2 + m2*L1^2;
aa(5)= L0*(m1*Lc1 + m2*L1);
aa(6)= I0 + m0*Lc0^2 + L0^2*(m1+m2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Initial Guess of System Constants %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

error = 1;                     % maximum error for constants

Lc0g = Lc0+2*error*Lc0*(rand-.5);
Lc1g = Lc1+2*error*Lc1*(rand-.5);
Lc2g = Lc2+2*error*Lc2*(rand-.5);

m0g = m0+2*error*m0*(rand-.5);
m1g = m1+2*error*m1*(rand-.5);
m2g = m2+2*error*m2*(rand-.5);

I0g = I0+2*error*I0*(rand-.5);
I1g = I1+2*error*I1*(rand-.5);
I2g = I2+2*error*I2*(rand-.5);

%%%%%%%%%%%%%%%%%%%%%%
% Initial Conditions %
%%%%%%%%%%%%%%%%%%%%%%

a0 = zeros(6,1);

a0(1) = I2g + m2g*Lc2g^2;
a0(2) = m2g*L1*Lc2g;
a0(3) = m2g*L0*Lc2g;
a0(4) = I1g + m1g*Lc1g^2 + m2g*L1^2;
a0(5) = L0*(m1g*Lc1g + m2g*L1);
% a0(6) = I0g + m0g*Lc0g^2 + L0^2*(m1g+m2g);
a0(6) = aa(6);

p0 = 100*eye(6);

t0 = 0;
tfinal = 15 ;
dt = .01;
x0 = [0.0; 0.0; 0.0; 0*pi/180; -55*pi/180; 15*pi/180];
```

```
tol = 1e-6;
trace = 0;

%%%%%%%%%%%%%%%%%%%%%%%%%
% Numerical Integration %
%%%%%%%%%%%%%%%%%%%%%%%%%

[t,x,thdd,u,a,h,hd,Ref,rxref,ryref] =
eul('peq',tfinal,dt,x0,a0,p0);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculate Momentum Wheel Speed & Tip Position %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Iw    = 0.0912;         % kg-m^2
AA    = [aa'];
thddw = zeros(1,length(t));
thdw  = zeros(1,length(t));
thdw(1) = 104.7;      % rad/sec (= 1000rpm)
rx = zeros(1,length(t));
ry = zeros(1,length(t));
rx(1) =
L0*cos(x(1,4))+L1*cos(sum(x(1,4:5)))+L2*cos(sum(x(1,4:6)));
ry(1) =
L0*sin(x(1,4))+L1*sin(sum(x(1,4:5)))+L2*sin(sum(x(1,4:6)));

for i=2:length(t);
  thddw(i) =-u(i,1)/Iw;
  thdw(i) = thddw(i)*(t(i)-t(i-1))+thdw(i-1);
  AA=[AA aa'];
  rx(i) =
L0*cos(x(i,4))+L1*cos(sum(x(i,4:5)))+L2*cos(sum(x(i,4:6)));
  ry(i) =
L0*sin(x(i,4))+L1*sin(sum(x(i,4:5)))+L2*sin(sum(x(i,4:6)));
end

%%%%%%%%%%%%%%%
% Plot Output %
%%%%%%%%%%%%%%%

% figplot
```

B. PEQ

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Equations of Motion for a Polynomial Reference Maneuver %
% peq.m                                                   %
% called by: ode2.m
% calls: ref.m, mgm.m, adap.m, angmo.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function  [xdot,thdd,U,A,P,H,Hd,Ref,rx,ry] = peq(t,x,a,p)

%%%%%%%%%%%%%%%%%
% Define States %
%%%%%%%%%%%%%%%%%
% x(1)=thd0
% x(2)=thd1
% x(3)=thd2
% x(4)=th0
% x(5)=th1
% x(6)=th2

%%%%%%%%%%%%%
% Constants %
%%%%%%%%%%%%%

L0 = .427;    % centerbody radius
L1 = .530;    % length of arm 1
L2 = .533;    % length of arm 2
L  = [L0;L1;L2];

Lc0 = .104;         % length to CM of centerbody
Lc1 = .403;         % length to CM arm 1
Lc2 = .314;         % length to CM arm 2
Lc  = [Lc0;Lc1;Lc2];

m0 = 65.96;          % centerbody mass
m1 = 2.34;           % arm 1 mass
m2 = 2.86;           % arm 2 mass
m  = [m0;m1;m2];

I0 = 5.74;        % centerbody inertia
I1 = .081;        % arm 1 inertia
I2 = .182;        % arm to inertia
I  = [I0;I1;I2];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Actual System Parameters %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%

aa(1)= I2 + m2*Lc2^2;
aa(2)= m2*L1*Lc2;
aa(3)= m2*L0*Lc2;
aa(4)= I1 + m1*Lc1^2 + m2*L1^2;
aa(5)= L0*(m1*Lc1 + m2*L1);
aa(6)= I0 + m0*Lc0^2 + L0^2*(m1+m2);


%%%%%%%%%%%%%%
% Controller %
%%%%%%%%%%%%%%

thd(1,1) = x(1);
thd(2,1) = x(2);
thd(3,1) = x(3);

th(1,1)  = x(4);
th(2,1)  = x(5);
th(3,1)  = x(6);

[MM,GM] = mgm(th,thd,a);

Kp=100*eye(3);
Kv=50*eye(3);

[Uref,thr,thdr,thddr,rx,ry] = ref(t,L,a);

Ref=[Uref;thr;thdr;thddr];
du=MM*(-Kv*(thd-thdr)-Kp*(th-thr));

Ul=MM*thddr+GM;

U=Uref+du;
% U=Ul+du;
% U=Uref;
%%%%%%%%%%%%%
% Plant EOM %
%%%%%%%%%%%%%

[MMa,GMa] = mgm(th,thd,aa);

thdd = inv(MMa)*(U-GMa);
xdot = [thdd;x(1);x(2);x(3)];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Adaptive Parameter Update %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[A,P,Phi,K] = adap(th,thd,thdd,a,U,p);
% test=MMa*thdd+GMa-Phi'*aa'
```

```
%%%%%%%%%%%%%%%%%%%%%%%
% Nonadaptive Control %
%%%%%%%%%%%%%%%%%%%%%%%

% A = a;
% P = p;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculate Angular Momentum %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[H,Hd] = angmo(m,I,L,Lc,th,thd,thdd);
```

C. REF

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Function to produce reference parameters %
% ref.m                                    %
% called by: peq.m                         %
% calls: mgm.m                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [Uref,thr,thdr,thddr,rx,ry] = ref(t,L,a);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Initial and Final Angles and Times %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

L0=L(1);
L1=L(2);
L2=L(3);

th0ri = 0;              % always=0
th1ri = -55*pi/180;
th2ri = 15*pi/180;

th0rf = 0;              % always=0
th1rf = 40*pi/180;
th2rf = 15*pi/180;

ths = 0;                % constant

t0 = 0;
tf = 10;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Initial And Final Vector Positons %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

r3x0 = L0*cos(ths)+L1*cos(ths+th1ri)+L2*cos(ths+th1ri+th2ri);
r3y0 = L0*sin(ths)+L1*sin(ths+th1ri)+L2*sin(ths+th1ri+th2ri);
r3xf = L0*cos(ths)+L1*cos(ths+th1rf)+L2*cos(ths+th1rf+th2rf);
r3yf = L0*sin(ths)+L1*sin(ths+th1rf)+L2*sin(ths+th1rf+th2rf);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Calculate Reference Maneuver %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

tau = ( t - t0) / ( tf - t0 );

f   = ( 10 * tau^3 - 15 * tau^4 + 6 * tau^5 );
fd  = ( 30 * tau^2 - 60 * tau^3 + 30 * tau^4);
f2d = (60*tau-180*tau^2+120*tau^3);
```

```
rx = r3x0 + ( r3xf - r3x0 ) * f;
ry = r3y0 + ( r3yf - r3y0 ) * f;

rxd = fd*( r3xf - r3x0 )/( tf - t0 );
ryd = fd*( r3yf - r3y0 )/( tf - t0 );

rxdd = f2d*(r3xf-r3x0)/((tf-t0)^2);
rydd = f2d*(r3yf-r3y0)/((tf-t0)^2);

if (t>tf);

  rx=r3xf;
  ry=r3yf;
  rxd=0;
  ryd=0;
  rxdd=0;
  rydd=0;

 end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Determine Inverse Kinematics %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Sx = L0*cos(ths);
Sy = L0*sin(ths);
SR = sqrt((rx-Sx)^2+(ry-Sy)^2);

B1 = atan2(ry-Sy,rx-Sx);
B2 = acos((L1^2+SR^2-L2^2)/(2*L1*SR));
B3 = acos((L1^2+L2^2-SR^2)/(2*L1*L2));

th1r = B1-B2-ths;
th2r = pi-B3;
thr  = [0;th1r;th2r];
% thr  = [th1r;th1r;th2r];

H(1,1) = -L2*sin(ths+th1r+th2r)-L1*sin(ths+th1r);
H(1,2) = -L2*sin(ths+th1r+th2r);
H(2,1) = L2*cos(ths+th1r+th2r)+L1*cos(ths+th1r);
H(2,2) = L2*cos(ths+th1r+th2r);

[thdr12] = inv(H)*[rxd;ryd];
thd1r = thdr12(1);
thd2r = thdr12(2);
thdr = [0;thd1r;thd2r];
% thdr = [thd1r;thd1r;thd2r];

Hd(1,1) =
-L2*(thd1r+thd2r)*cos(ths+th1r+th2r)-L1*thd1r*cos(ths+th1r);
```

```
Hd(1,2) = -L2*(thd1r+thd2r)*cos(ths+th1r+th2r);
Hd(2,1) =
-L2*(thd1r+thd2r)*sin(ths+th1r+th2r)-L1*thd1r*sin(ths+th1r);
Hd(2,2) = -L2*(thd1r+thd2r)*sin(ths+th1r+th2r);

[thddr12] = inv(H)*([rxdd;rydd]-Hd*[thd1r;thd2r]);
thdd1r = thddr12(1);
thdd2r = thddr12(2);
thddr = [0;thdd1r;thdd2r];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculate Reference Control Torques %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[MMr,GMr] = mgm(thr,thdr,a);

Uref  =  MMr*thddr+GMr;
```

D. EUL

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
% Discrete Euler Integration
    %
% eul.m
    %
% called by:pcont.m
    %
% calls: peq.m
    %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
function
[tout,xout,thddout,uout,aout,Hout,Hdout,Refout,rxout,ryout]=
...
                          eul(FunFcn,tf,dt,x0,a0,p0)


%%%%%%%%%%%%%%%%%%
% Initialization %
%%%%%%%%%%%%%%%%%%

t = 0;
x = x0(:);
u = zeros(3,1);
a = a0(:);
thdd = zeros(3,1);
p = p0;
H = 0;
Hd = 0;
Ref=zeros(12,1);
Ref(4:6)=[0;-55;15]*pi/180;
L0 = .427;
L1 = .530;
L2 = .533;

tout = [];
xout = [];
thddout = [];
uout = [];
aout = [];
Hout = [];
Hdout = [];
Refout = [];
rxout = [];
ryout = [];

%%%%%%%%%%%%%%%%%
```

```
% The main loop %
%%%%%%%%%%%%%%%%%

for i=1:(tf/dt)+1

[xd,thdd,u,A,P,H,Hd,Ref,rx,ry] = feval(FunFcn,t,x,a,p);

tout = [tout; t];
xout = [xout; x.'];
thddout = [thddout; thdd.'];
uout = [uout; u.'];
aout = [aout; a.'];
Hout = [Hout; H];
Hdout = [Hdout; Hd];
Refout = [Refout; Ref.'];
rxout = [rxout;rx];
ryout = [ryout;ry];

t = t + dt;
x = x + xd*dt;
a = A;
p = P;

end
```

E. MGM

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Function to Calculate 'M' matrix and 'G' vector %
% mgm.m                                           %
% called by: peq.m, ref.m                         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [MM,GM] = mgm(th,thd,a);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Define Angles & Angular Rates %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

th0 = th(1);
th1 = th(2);
th2 = th(3);
thd0 = thd(1);
thd1 = thd(2);
thd2 = thd(3);

%%%%%%%%%%%%%%%%%%%%%%%%
% Calculate 'M' matrix %
%%%%%%%%%%%%%%%%%%%%%%%%

MM(3,3)  = a(1);
MM(2,3)  = MM(3,3)+a(2)*cos(th2);
MM(3,2)  = MM(2,3);
MM(1,3)  = MM(2,3)+a(3)*cos(th1+th2);
MM(3,1)  = MM(1,3);
MM(2,2)  = MM(2,3)+a(2)*cos(th2) + a(4);
MM(1,2)  = MM(2,2)+a(3)*cos(th1+th2) + a(5)*cos(th1);
MM(2,1)  = MM(1,2);
MM(1,1)  = MM(2,2)+2*a(3)*cos(th1+th2)+2*a(5)*cos(th1)+a(6);

%%%%%%%%%%%%%%%%%%%%%%%%
% Calculate 'G' vector %
%%%%%%%%%%%%%%%%%%%%%%%%

ca2 = thd2*(2*thd0+2*thd1+thd2)*sin(th2);

GM(1,1) = -a(5)*(thd1^2+2*thd0*thd1)*sin(th1)-a(2)*ca2...

-a(3)*(2*thd0*(thd1+thd2)+(thd1+thd2)^2)*sin(th1+th2);
G      M     (      2      ,      1      )                  =
a(5)*thd0^2*sin(th1)-a(2)*ca2+a(3)*thd0^2*sin(th1+th2);
G      M     (      3      ,      1      )                  =
a(2)*(thd1+thd2)^2*sin(th2)+a(3)*thd0^2*sin(th1+th2);
```

F. ANGMO

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Subroutine to calculate system angular nmomentum %
% angmo.m
% called by: peq.m                                 %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [H,Hd]=angmo(m,I,L,Lc,th,thd,thdd);

%%%%%%%%%%%%%%%%%%%%%%%%
% Local variable names %
%%%%%%%%%%%%%%%%%%%%%%%%

m0 = m(1);
m1 = m(2);
m2 = m(3);

I0 = I(1);
I1 = I(2);
I2 = I(3);

L0 = L(1);
L1 = L(2);
L2 = L(3);

Lc0 = Lc(1);
Lc1 = Lc(2);
Lc2 = Lc(3);

th0=th(1);
th1=th(2);
th2=th(3);

thd0=thd(1);
thd1=thd(2);
thd2=thd(3);

thdd0=thdd(1);
thdd1=thdd(2);
thdd2=thdd(3);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Angular Momentum Equations %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

H0 = thd0*(I0+m0*Lc0^2);
H1 = thd0*(I1+m1*(L0^2+Lc1^2+2*L0*Lc1*cos(th1)))...
    +thd1*(I1+m1*(Lc1^2+L0*Lc1*cos(th1)));
H2 = thd0*(I2+m2*(L0^2+L1^2+Lc2^2+2*L0*L1*cos(th1)...
           +2*L1*Lc2*cos(th2)+2*L0*Lc2*cos(th1+th2)))...
    +thd1*(I2+m2*(L1^2+Lc2^2+L0*L1*cos(th1)...
           +2*L1*Lc2*cos(th2)+L0*Lc2*cos(th1+th2)))...
    +thd2*(I2+m2*(Lc2^2+L1*Lc2*cos(th2)+L0*Lc2*cos(th1+th2)));

H=H0+H1+H2;

%%%%%%%%%%%%%%%%%%
% Hdot Equations %
%%%%%%%%%%%%%%%%%%

Hd0 = thdd0*(I0+m0*Lc0^2);
Hd1 = thdd0*(I1+m1*(L0^2+Lc1^2+2*L0*Lc1*cos(th1)))...
     +thdd1*(I1+m1*(Lc1^2+L0*Lc1*cos(th1)))...
     -thd0*thd1*2*m1*L0*Lc1*sin(th1)...
     -thd1^2*m1*L0*Lc1*sin(th1);
Hd2 = thdd0*(I2+m2*(L0^2+L1^2+Lc2^2+2*L0*L1*cos(th1)...
           +2*L1*Lc2*cos(th2)+2*L0*Lc2*cos(th1+th2)))...
     +thdd1*(I2+m2*(L1^2+Lc2^2+L0*L1*cos(th1)...
           +2*L1*Lc2*cos(th2)+L0*Lc2*cos(th1+th2)))...

+thdd2*(I2+m2*(Lc2^2+L1*Lc2*cos(th2)+L0*Lc2*cos(th1+th2)))...
     -thd0*thd1*2*m2*(L0*L1*sin(th1)+L0*Lc2*sin(th1+th2))...
     -thd0*thd2*2*m2*(L1*Lc2*sin(th2)+L0*Lc2*sin(th1+th2))...
     -thd1*thd2*2*m2*(L1*Lc2*sin(th2)+L0*Lc2*sin(th1+th2))...
     -thd1^2*m2*(L0*L1*sin(th1)+L0*Lc2*sin(th1+th2))...
     -thd2^2*m2*(L1*Lc2*sin(th2)+L0*Lc2*sin(th1+th2));

Hd=Hd0+Hd1+Hd2;
```

G. ADAP

```
%%%%%%%%%%%%%%%%%%%%%%%%%
% Adaption Law          %
% A=adap(th,thd,thdd,a) %
% called by: peq.m      %
%%%%%%%%%%%%%%%%%%%%%%%%%

function [A,P,Phi,K]=adap(th,thd,thdd,a,U,p);

%%%%%%%%%%%%%%%%%%%%%%%%
% Local Variable Names %
%%%%%%%%%%%%%%%%%%%%%%%%

th0=th(1);
th1=th(2);
th2=th(3);

thd0=thd(1);
thd1=thd(2);
thd2=thd(3);

thdd0=thdd(1);
thdd1=thdd(2);
thdd2=thdd(3);

%%%%%%%%%%%%%%
% Phi Matrix %
%%%%%%%%%%%%%%

Phi(1,1) = thdd0+thdd1+thdd2;
Phi(1,2) = Phi(1,1);
Phi(1,3) = Phi(1,1);
Phi(2,1) = (2*thdd0+2*thdd1+thdd2)*cos(th2)...
          -thd2*(2*thd0+2*thd1+thd2)*sin(th2);
Phi(2,2) = Phi(2,1);
Phi(2,3) = (thdd0+thdd1)*cos(th2) + (thd1+thd2)^2*sin(th2);
Phi(3,1) = (2*thdd0+thdd1+thdd2)*cos(th1+th2)...
          -(2*thd0*(thd1+thd2) + (thd1+thd2)^2)*sin(th1+th2);
Phi(3,2) = thdd0*cos(th1+th2) + thd0^2*sin(th1+th2);
Phi(3,3) = Phi(3,2);
Phi(4,1) = thdd0+thdd1;
Phi(4,2) = Phi(4,1);
Phi(4,3) = 0;
Phi(5,1) = (2*thdd0+thdd1)*cos(th1) -
(thd1^2+2*thd0*thd1)*sin(th1);
Phi(5,2) = thdd0*cos(th1) + thd0^2*sin(th1);
Phi(5,3) = 0;
Phi(6,1) = thdd0;
Phi(6,2) = 0;
Phi(6,3) = 0;

%%%%%%%%%%%%%%%%%%%%%%
% Adaption Equations %
%%%%%%%%%%%%%%%%%%%%%%

K = p*Phi*inv(eye(3)+Phi'*p*Phi);

% A=a;
A = a+K*(U-Phi'*a);

% P = eye(6);
P = p-K*Phi'*p+eye(6);
```

APPENDIX C: EXPERIMENTAL CONTROL BLOCK DIAGRAMS

This appendix includes the diagrams of the system build block diagrams made to control the SRS. Both is the parent superblock. The others are lower level superblocks.

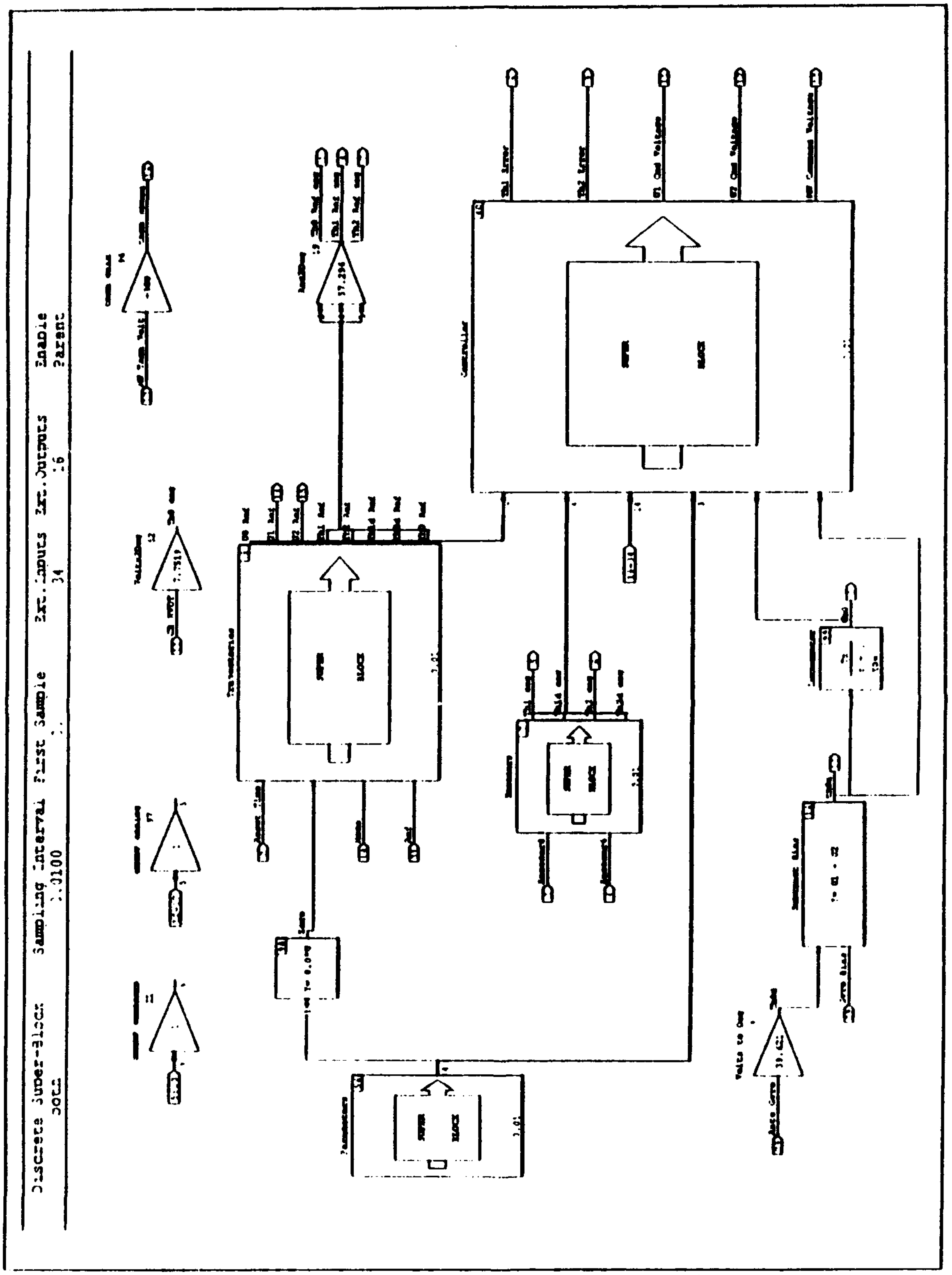

Figure 41: Overall Control Block Diagram

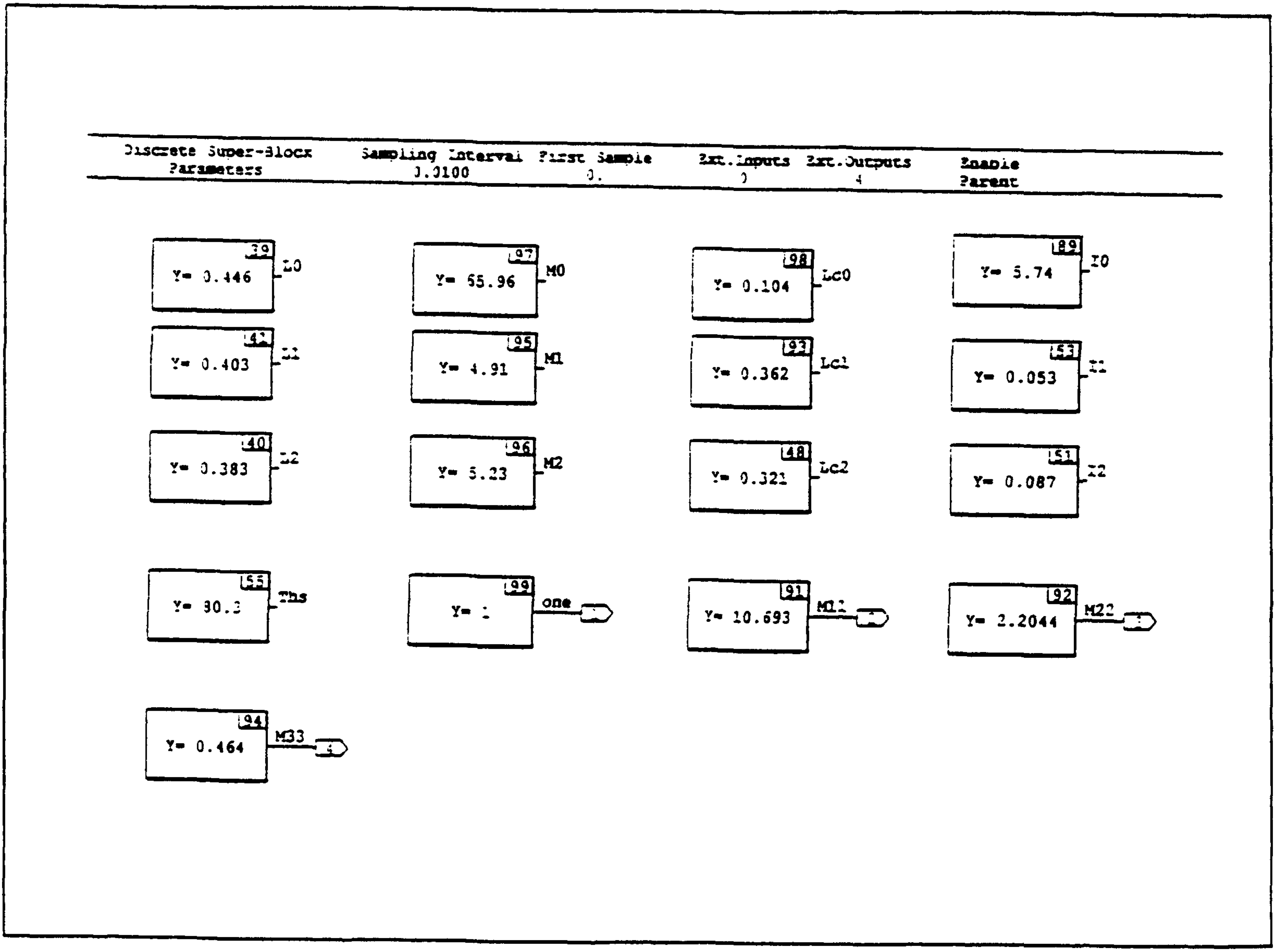

Figure 42: Parameters Block Diagram

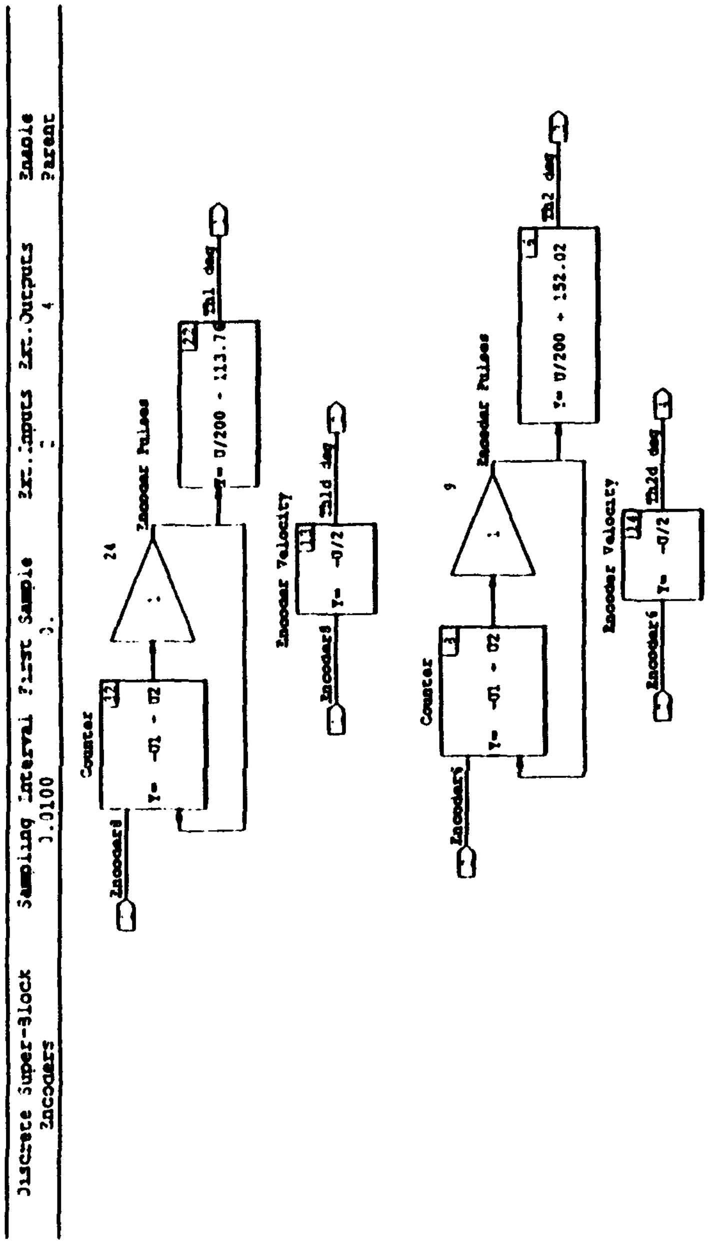

Figure 43: Encoders Block Diagram

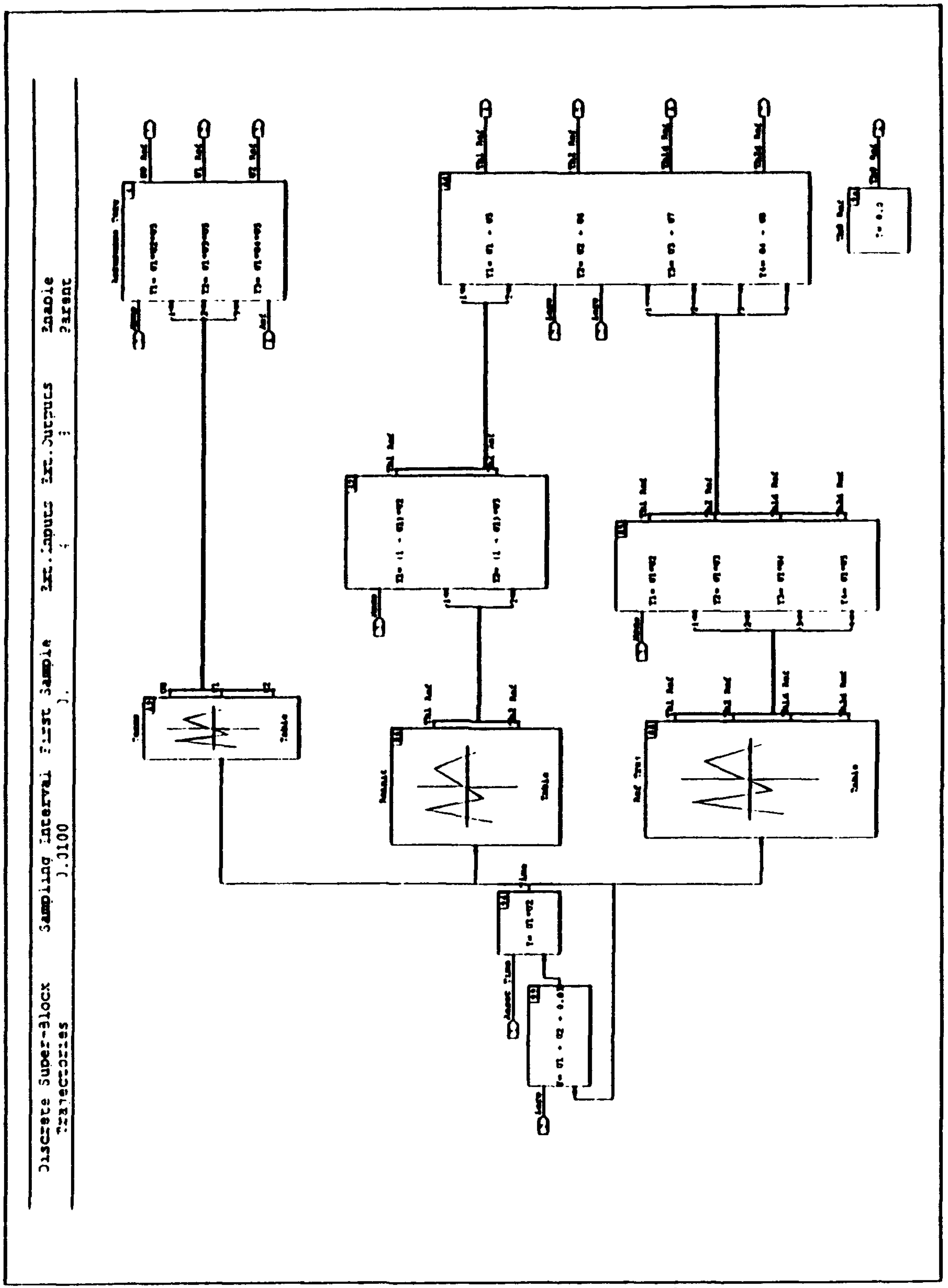

Figure 44: Trajectories Block Diagram

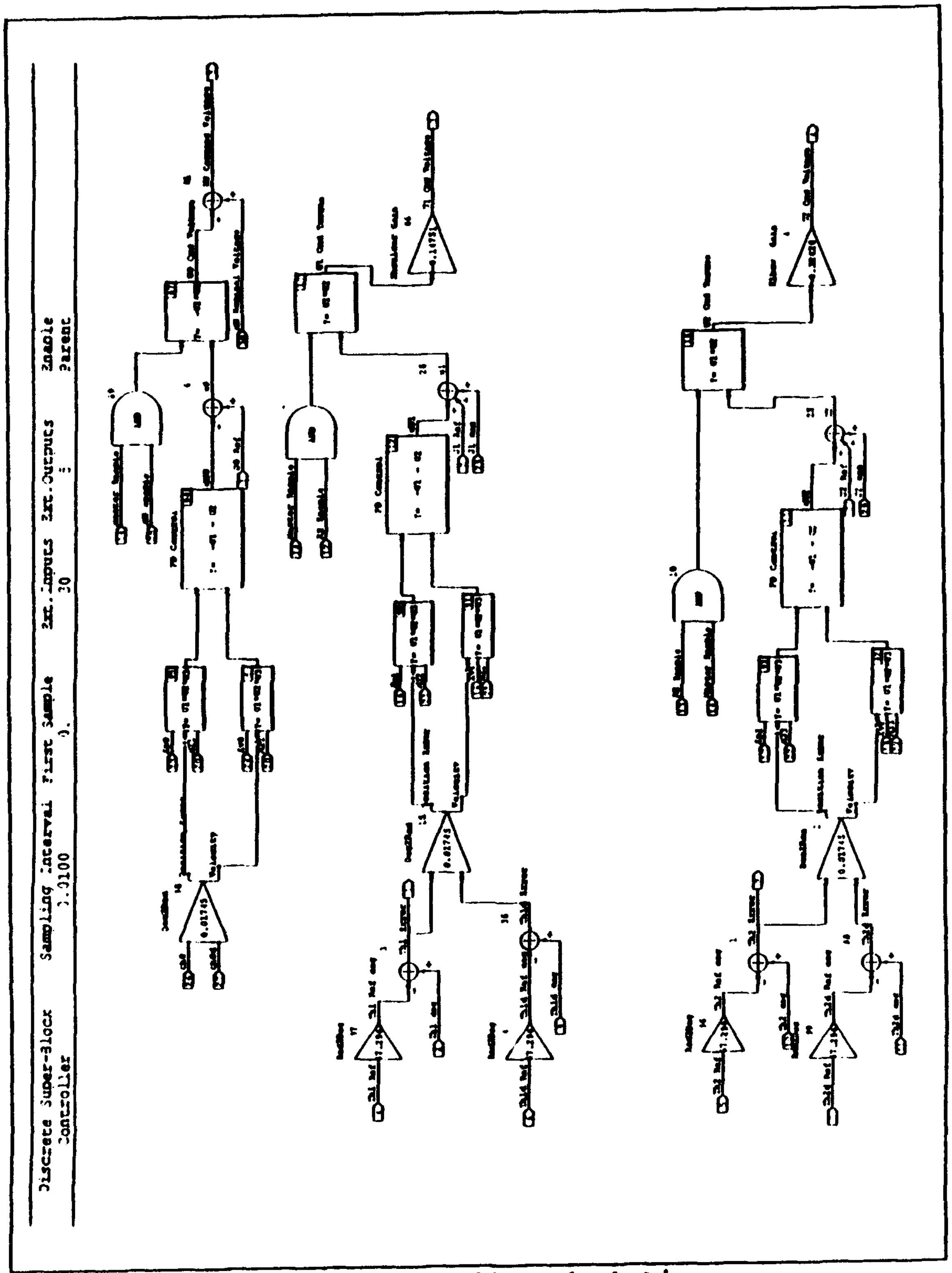

Figure 45: Controller Block Diagram

LIST OF REFERENCES

1. Dubosky, S. and DesForges, D., "The Application of Model-Reference Adaptive Control to Robotic Manipulators," J. Dyn. Sys. Meas., and Cont., vol 101, pp.193-200, 1979.

2. Horowitz, R. and Tomizuka M., "An Adaptive Control Scheme for Mechanical Manipulators - Compensation of Nonlinearity and Decoupling Control," ASME paper 80 WA/DSC-6, 1980.

3. Koivo, A., "Controll of Robotic Manipulator with Adaptive Controller," IEEE Conf. Dec. and Cont., San Diego California, 1981.

4. Balestrino, A., De Maria G., and Sciavicco L., "Adaptive Control of Robotic Manipulators," AFCET Congres Automatique, Nantes, France, 1981.

5. Nicolo F. and Katende, IASTED Conf. Robotics and Automation, Lugano, 1982.

6. Atkeson, C.G., An, C.H., and Hollerbach,J.M., "Estimation of Inertial Parameters of Rigid Body Links of Manipulators," IEEE Dec. and Cont., Fort Lauderdale Florida, 1985.

7. Khosla P. ang Kanade T., "Parameter Identification of Robot Dynamics," IEEE Conf. Dec. and Cont., Fort Lauderdale Florida, 1985.

8. Sorenson, Dennis, *Design and Control of a Space Based Two-Link Manipulator with Lyapunov Based Control Laws,* Master's Thesis, Naval Postgraduate School, Monterey California, September 1992.

9. Yale, Gary E., *Cooperative Control of Multiple Space Manipulators,* Docterate Dissertation, Naval Postgraduate School, Monterey California, September 1993.

10. Watkins Jr., R.J., *The Attitude Control of Flexible Spacecraft,* Master's Thesis, Naval Postgraduate School, Monterey California, June 1991.

11. Hailey, Jefferey A., *Experimental Verification of Attitude Control Techniques for Flexible Spacecraft Slew Maneuvers,* Master's Thesis, Naval Postgraduate School, Monterey California, March 1992.

INITIAL DISTRIBUTION LIST

No. Copies

1. Defense Technical Information Center 2
   Cameron Station
   Alexandria VA 22304-6145

2. Library, Code 52 2
   Naval Postgraduate School
   Monterey CA 93943-5101

3. Chairman, Code AA 1
   Department of Aeronautics and Astronautics
   Naval Postgraduate School
   Montertey CA 93943-5002

4. Chairman, Code SP 1
   Space Systems Academic Group
   Naval Postgraduate School
   Montertey CA 93943-5002

5. Chairman, Code EC 1
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Montertey CA 93943-5121

6. Professor Brij N. Agrawal, Code AA/Ag 1
   Department of Aeronautics and Astronautics
   Naval Postgraduate School
   Montertey CA 93943-5002

7. Professor Roberto Cristi, Code EC/Cx 1
   Department of Aeronautics and Astronautics
   Naval Postgraduate School
   Montertey CA 93943-5121

8. Professor Hyochoong Bang, Code AA/Bn 1
   Department of Aeronautics and Astronautics
   Naval Postgraduate School
   Montertey CA 93943-5002

9. LT George Janvier 1
   HSL-41
   Naval Air Station North Island
   San Diego CA 92135-7126